

# **UNIVERSIDAD CARLOS III DE MADRID**

## **ESCUELA POLITÉCNICA SUPERIOR**

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA**  
**LABORATORIO DE SISTEMAS INTELIGENTES**



# **ALIMENTACIÓN ELÉCTRICA DE UN VEHÍCULO INTELIGENTE: DISEÑO E IMPLEMENTACIÓN**

**PROYECTO FIN DE CARRERA**  
**INGENIERÍA TÉCNICA INDUSTRIAL: ELECTRÓNICA**

Autor: María Jesús Gutiérrez Hernández  
Tutor: Basam Musleh Lancis

Leganés, Enero 2012



*A mi madre quien me enseñó el camino  
y a Luis que me ayudó a encontrar la salida del laberinto.*

*Gracias a mi padre y a mi hermana por  
no decirme, nunca, que abandonara, a mis  
abuelos por su inestimable ayuda paciencia  
y a todos aquellos que habéis colaborado  
para que este día llegara.*



# 0 Índice de contenidos

<b>0</b>	<b>ÍNDICE DE CONTENIDOS</b>	<b>5</b>
<b>1</b>	<b>INTRODUCCIÓN</b>	<b>11</b>
<b>1.1</b>	<b>POSICIÓN EUROPEA</b>	<b>11</b>
<b>1.2</b>	<b>VEHÍCULOS INTELIGENTES</b>	<b>13</b>
1.2.1	CARACTERÍSTICAS DE LOS VEHÍCULOS INTELIGENTES	13
1.2.2	SISTEMA ANTI BLOQUEO DE FRENOS (ABS)	13
1.2.3	CONTROL DE CRUCERO ADAPTATIVO (ACC)	14
1.2.4	LUCES AUTOADAPTABLES	14
1.2.5	ASISTENTE DE CAMBIO DE CARRIL.	14
1.2.6	DETECCIÓN DEL ÁNGULO MUERTO	14
1.2.7	SISTEMAS DE CONTROL Y ALERTA DE SOMNOLENCIA	15
1.2.8	SISTEMA DE LLAMADA DE EMERGENCIA DEL VEHÍCULO (eCALL)	15
1.2.9	SISTEMA ELECTRÓNICO DE ASISTENCIA A LA FRENADA (EBS, <i>ELECTRONIC BRAKE ASSIST SYSTEM</i> )	15
1.2.10	CONTROL ELECTRÓNICO DE ESTABILIDAD (ESC, <i>ELECTRONIC BRAKE ASSIST SYSTEM</i> )	15
1.2.11	SISTEMA DE AVISO DE CAMBIO INVOLUNTARIO DE CARRIL (LDWS)	16
1.2.12	VISIÓN NOCTURNA	16
1.2.13	SISTEMAS DE PROTECCIÓN DE PEATONES Y USUARIOS VULNERABLES EN LA VÍA PÚBLICA	17
1.2.14	ALERTA DE VELOCIDAD EXCESIVA	17
<b>1.3</b>	<b>¿EXISTE UNA NECESIDAD REAL DE ESTOS SISTEMAS?</b>	<b>17</b>
1.3.1	PROBLEMAS DE IMPLANTACIÓN	17
1.3.2	VENTAJAS DE IMPLANTACIÓN	18
1.3.3	POTENCIAL DE LOS SISTEMAS INTELIGENTES	18
<b>2</b>	<b>INTELLIGENT VEHICLE BASED ON VISUAL INFORMATION</b>	<b>19</b>
<b>2.1</b>	<b>VEHÍCULO IVVI</b>	<b>19</b>
<b>2.2</b>	<b>SISTEMAS QUE INCORPORA EL IVVI</b>	<b>19</b>
<b>3</b>	<b>SISTEMA DE GESTIÓN DE BATERÍA</b>	<b>21</b>
<b>3.1</b>	<b>DISEÑO</b>	<b>21</b>
3.1.1	LA IDEA	21
3.1.2	LAS ESPECIFICACIONES	21
3.1.3	ARQUITECTURA DEL DISEÑO	22
<b>4</b>	<b>BATERIA</b>	<b>23</b>

<b>4.1</b>	<b>INTRODUCCIÓN</b>	<b>23</b>
<b>4.2</b>	<b>ELEMENTOS DE UNA BATERÍA COMERCIAL DE PLOMO-ÁCIDO</b>	<b>23</b>
<b>4.3</b>	<b>FUNCIONAMIENTO DE UNA BATERÍA COMERCIAL DE PLOMO-ÁCIDO</b>	<b>24</b>
4.3.1	PROCESO DE CARGA Y DESCARGA	25
4.3.2	INFLUENCIA DE LA TEMPERATURA EN LA BATERÍA	25
<b>4.4</b>	<b>LA CAPACIDAD EN LA BATERÍA</b>	<b>26</b>
4.4.1	TIEMPO DE DURACIÓN DE UNA BATERÍA	27
4.4.2	VARIACIÓN DE LA CAPACIDAD DE LA BATERÍA CON LA TEMPERATURA	27
4.4.3	ESTIMACIÓN DE LA CAPACIDAD SEGÚN EL VOLTAJE	28
<b>4.5</b>	<b>LA BATERÍA ADICIONAL</b>	<b>28</b>
<b>5</b>	<b>SENSOR DETECTOR DE RED</b>	<b>29</b>
<b>5.1</b>	<b>DISEÑO DE UN SENSOR DETECTOR DE RED</b>	<b>29</b>
5.1.1	OBJETIVO	29
<b>5.2</b>	<b>DIAGRAMA DE BLOQUES DEL SENSORED</b>	<b>29</b>
<b>5.3</b>	<b>RESTRICCIONES QUE CONDICIONAN EL DISEÑO</b>	<b>30</b>
<b>5.4</b>	<b>DESARROLLO</b>	<b>30</b>
<b>5.5</b>	<b>ACONDICIONAMIENTO DE LA ENTRADA</b>	<b>31</b>
5.5.1	DIAGRAMA DE BLOQUES	31
<b>5.6</b>	<b>BLOQUE DE REDUCCIÓN DE LA SEÑAL DE ENTRADA</b>	<b>31</b>
5.6.1	SIMULACIÓN EN ORCAD DE UN TRANSFORMADOR	31
5.6.2	SELECCIÓN DEL TRANSFORMADOR	33
<b>5.7</b>	<b>BLOQUE DE RECTIFICACIÓN</b>	<b>35</b>
<b>5.8</b>	<b>BLOQUE DE FILTRADO</b>	<b>35</b>
<b>5.9</b>	<b>BLOQUE DE COMPARACIÓN</b>	<b>39</b>
5.9.1	SELECCIÓN DEL AMPLIFICADOR OPERACIONAL	39
<b>5.10</b>	<b>SELECCIÓN DE LA TENSIÓN DE REFERENCIA</b>	<b>39</b>
<b>5.11</b>	<b>SELECCIÓN DE LA TENSIÓN DE ALIMENTACIÓN DEL OPERACIONAL</b>	<b>41</b>
<b>5.12</b>	<b>CIRCUITO ELECTRICO FINAL</b>	<b>41</b>
<b>5.13</b>	<b>ACONDICIONAMIENTO A LA TARJETA TS-7200</b>	<b>42</b>
<b>6</b>	<b>SENSOR DE CORRIENTE</b>	<b>43</b>
<b>6.1</b>	<b>OBJETIVO</b>	<b>43</b>
<b>6.2</b>	<b>RESTRICCIONES</b>	<b>43</b>
<b>6.3</b>	<b>MEDIDA DE LA CORRIENTE</b>	<b>43</b>
<b>6.4</b>	<b>TRANSDUCTORES DE CORRIENTE BASADOS EN EL EFECTO HALL</b>	<b>43</b>
<b>6.5</b>	<b>SELECCIÓN DEL TRANSDUCTOR DE CORRIENTE</b>	<b>45</b>
<b>6.6</b>	<b>TRANSDUCTOR SELECCIONADO</b>	<b>46</b>
6.6.1	CARACTERÍSTICAS QUE DETERMINAN EL TRANSDUCTOR SELECCIONADO	46
<b>6.7</b>	<b>BLOQUE ACONDICIONADOR</b>	<b>48</b>
6.7.1	GANANCIA DEL AMPLIFICADOR DE INSTRUMENTACIÓN	48
6.7.2	RELACIÓN ENTRE LA SALIDA DEL TRANSDUCTOR Y EL ACONDICIONAMIENTO	51
<b>6.8</b>	<b>ADAPTACIÓN DEL SENSOR A LA PLACA TS-7200</b>	<b>54</b>
<b>7</b>	<b>ESTIMACIÓN DE LA CAPACIDAD DE UNA BATERÍA</b>	<b>55</b>
<b>7.1</b>	<b>CIRCUITO MEDIDOR DE LA TENSIÓN EN BORNES DE LA BATERÍA</b>	<b>55</b>
<b>7.2</b>	<b>ACONDICIONAMIENTO A LA ENTRADA DE LA TARJETA TS-7200</b>	<b>57</b>
<b>8</b>	<b>EL SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA</b>	<b>59</b>

<b>8.1</b>	<b>SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA (SAI)</b>	<b>59</b>
<b>8.2</b>	<b>PARÁMETROS DE UN SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA</b>	<b>59</b>
<b>8.3</b>	<b>TIPOS DE SISTEMAS DE ALIMENTACIÓN ININTERRUMPIDA</b>	<b>59</b>
<b>8.4</b>	<b>SELECCIÓN DEL SISTEMAS DE ALIMENTACIÓN ININTERRUMPIDA</b>	<b>60</b>
8.4.1	CÁLCULO DE LA POTENCIA EN VATIOS	60
8.4.2	ELECCIÓN DEL SAI	61
8.4.3	PUESTA EN MARCHA	61
<b>9</b>	<b>SISTEMA DE CONVERSIÓN DE LA ALIMENTACIÓN</b>	<b>63</b>
<b>9.1</b>	<b>INVERSOR ELÉCTRICO</b>	<b>63</b>
<b>9.2</b>	<b>SELECCIÓN DEL INVERSOR</b>	<b>63</b>
<b>9.3</b>	<b>INTEGRACIÓN DEL INVERSOR</b>	<b>64</b>
<b>10</b>	<b>TS-7200</b>	<b>65</b>
<b>10.1</b>	<b>INTRODUCCIÓN A LA TARJETA</b>	<b>65</b>
10.1.1	TS-7200	65
10.1.2	SERIES Y MODELOS	65
10.1.3	SERIE TS-72XX	65
<b>10.2</b>	<b>DESCRIPCIÓN</b>	<b>66</b>
<b>10.3</b>	<b>INTERFACES</b>	<b>67</b>
10.3.1	ENTRADAS Y SALIDAS DIGITALES	67
10.3.2	PUERTO SERIE	67
10.3.3	CONVERSOR A/D	67
<b>10.4</b>	<b>COMPONENTES HARDWARE</b>	<b>68</b>
10.4.1	PROCESADOR	68
10.4.2	MEMORIA	69
10.4.3	GLUE LOGIC CPLD	69
10.4.4	WATCHDOG TIMER (WDT)	69
<b>10.5</b>	<b>CABEZAL JTAG (JOINT TEST ACTION GROUP)</b>	<b>70</b>
<b>10.6</b>	<b>CABEZAL LCD</b>	<b>70</b>
<b>10.7</b>	<b>PC/104 BUS EXPANSIÓN</b>	<b>71</b>
<b>10.8</b>	<b>CONECTORES</b>	<b>71</b>
10.8.1	PUERTO ETHERNET	71
10.8.2	CONECTOR USB	72
10.8.3	CONECTOR PARA MEMORIA SCF	72
10.8.4	CONECTOR COM1	72
10.8.5	COM2	73
<b>11</b>	<b>PROGRAMACIÓN DE LA TARJETA TS-7200</b>	<b>75</b>
<b>11.1</b>	<b>REPRESENTACIÓN EN C DE LA INFORMACIÓN</b>	<b>75</b>
<b>11.2</b>	<b>MÁSCARAS</b>	<b>75</b>
11.2.1	FORZAR BITS A 0	76
11.2.2	DETECTAR EL ESTADO DE UN BIT	76
11.2.3	FORZAR BITS A 1	77
11.2.4	COMPLEMENTAR BITS	77
11.2.5	DESPLAZAMIENTOS	78
11.2.6	APLICACIÓN DE LAS MÁSCARAS A LA TARJETA TS-7200	79
<b>11.3</b>	<b>TS-7200 Y LINUX</b>	<b>80</b>
11.3.1	SISTEMA OPERATIVO TS-LINUX	80
11.3.2	PRINCIPALES COMANDOS BUSYBOX	80

<b><u>12</u></b>	<b><u>CABEZAL DIO1</u></b>	<b><u>81</u></b>
12.1	REGISTROS DR Y DDR	81
12.2	INTERRUPCIONES EN LAS DIO	82
12.3	MAPA DE MEMORIA DEL CABEZAL DIO1	82
12.4	GPIO (GENERAL PURPOSE INPUT/OUTPUT)	82
12.5	EL PUERTO B	84
12.5.1	PBDDR	85
12.5.2	PBDR	86
12.6	INTERFAZ SPI	86
12.7	MATRIZ DE TECLADO	86
<b><u>13</u></b>	<b><u>CONVERSOR ANALÓGICO DIGITAL</u></b>	<b><u>87</u></b>
13.1	CARACTERÍSTICAS DEL CONVERSOR A/D EP9302 DE CIRRUS	87
13.2	INTERPRETACIÓN DEL CONVERTIDOR ANALÓGICO DIGITAL A EP9302 DE CIRRUS	88
13.3	CONFIGURACIÓN SOFTWARE	89
13.4	LEDs JUMPERS Y BOTONES	91
13.4.1	LEDs	91
13.4.2	JUMPERS	92
13.4.3	BOTONES	93
<b><u>14</u></b>	<b><u>PUESTA EN MARCHA DE LA TARJETA</u></b>	<b><u>95</u></b>
14.1	ALIMENTACIÓN DE LA TARJETA TS-7200	95
14.2	CONEXIONES	95
<b><u>15</u></b>	<b><u>ENTORNOS DE DESARROLLO</u></b>	<b><u>97</u></b>
15.1	ECLIPSE	97
15.1.1	ENTORNO ECLIPSE. CONFIGURACIÓN DE LA VENTANA DE TRABAJO	98
15.1.2	CREAR UN WORKSPACE NUEVO O CAMBIAR DE WORKSPACE	100
15.1.3	CREACIÓN DE UN PROYECTO EN ECLIPSE	100
15.1.4	COMPILACIÓN	108
15.1.5	INCLUIR LIBRERÍAS EN EL PROYECTO	111
15.2	FILEZILLA	112
15.2.1	ACCESO AL PROGRAMA	112
15.2.2	ENTORNO	112
15.3	CONEXIÓN A LA TARJETA TS-7200	114
15.4	TRANSFERENCIA DE ARCHIVOS	117
15.4.1	PROBLEMAS EN LA COMUNICACIÓN	118
15.4.2	FINALIZACIÓN DE LA COMUNICACIÓN MEDIANTE FILEZILLA	118
<b><u>16</u></b>	<b><u>COMUNICACIÓN</u></b>	<b><u>119</u></b>
16.1	INTRODUCCIÓN A LOS SOCKETS	119
16.2	SOCKET EN C	120
16.2.1	CREACIÓN DE UN SOCKET	120
16.2.2	FUNCIÓN <i>BIND()</i>	120
16.2.3	FUNCIÓN <i>LISTEN()</i> Y FUNCIÓN <i>ACCEPT()</i>	121
16.2.4	ENVIAR Y RECIBIR DATOS	121

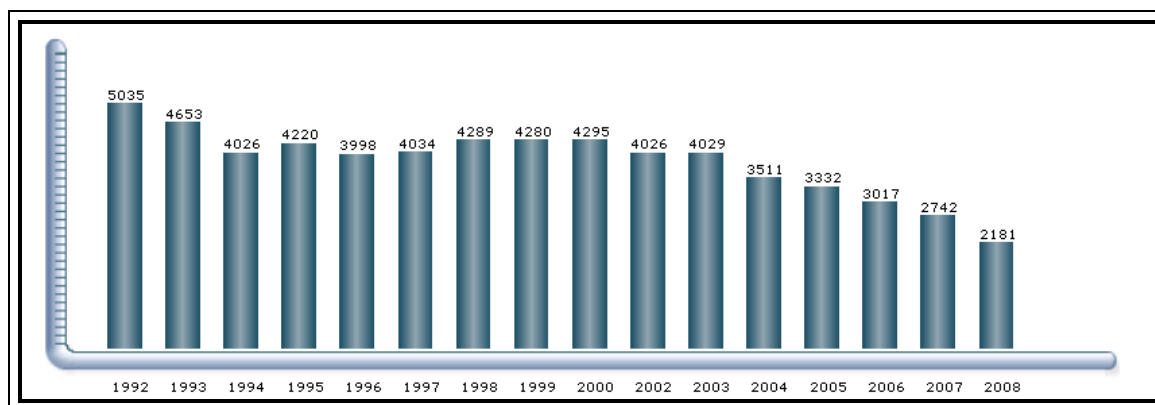


16.2.5	FINALIZAR LA CONEXIÓN	121
<b>16.3</b>	<b>SOCKET EN PYTHON</b>	<b>121</b>
<b>16.4</b>	<b>SECUENCIA DE EVENTOS EN LA COMUNICACIÓN</b>	<b>122</b>
<b><u>17</u></b>	<b><u>EASYBATTERY</u></b>	<b><u>123</u></b>
<b>17.1</b>	<b>ACCESO AL PROGRAMA EASYBATTERY</b>	<b>123</b>
<b>17.2</b>	<b>ENTORNO</b>	<b>123</b>
17.2.1	ESTABLECER CONEXIÓN	123
17.2.2	CERRAR COMUNICACIÓN	125
<b>17.3</b>	<b>BOTÓN COMPROBAR CONEXIÓN A LA RED</b>	<b>125</b>
<b>17.4</b>	<b>BOTÓN COMPROBAR INTENSIDAD DEMANDADA</b>	<b>126</b>
17.4.1	MODIFICACIÓN DE LA CAPACIDAD DE LA BATERÍA ACTUAL	126
17.4.2	TABLA DE DATOS	127
17.4.3	ELIMINAR	127
<b>17.5</b>	<b>PESTAÑA IMPRESIÓN DE DATOS</b>	<b>128</b>
17.5.1	GUARDAR UN DOCUMENTO	128
17.5.2	VISTA PREVIA DE UN DOCUMENTO	128
17.5.3	CONFIGURAR UN DOCUMENTO	129
17.5.4	IMPRIMIR UN DOCUMENTO	129
<b>17.6</b>	<b>CAPACIDAD ESTIMADA</b>	<b>129</b>
<b>17.7</b>	<b>CONFIGURACIÓN DE LA VENTANA PRINCIPAL</b>	<b>130</b>
17.7.1	FONDO	130
17.7.2	FECHA Y HORA	131
<b>17.8</b>	<b>AYUDA</b>	<b>131</b>
17.8.1	ACERCA DE EASYBATTERY	131
<b><u>18</u></b>	<b><u>PRESUPUESTO</u></b>	<b><u>133</u></b>
<b>18.1</b>	<b>DESGLOSE DE COSTES</b>	<b>133</b>
<b><u>19</u></b>	<b><u>TRABAJOS FUTUROS</u></b>	<b><u>135</u></b>
<b><u>20</u></b>	<b><u>APÉNDICES</u></b>	<b><u>137</u></b>
<b>20.1</b>	<b>APÉNDICE A: ESQUEMÁTICOS TS-7200</b>	<b>139</b>
<b>20.2</b>	<b>APÉNDICE B: MAPA DE MEMORIA</b>	<b>142</b>
<b>20.3</b>	<b>APÉNDICE C: PROGRAMAS</b>	<b>144</b>
20.3.1	PROGRAMA EASYBATTERY	144
20.3.2	PROGRAMA TS-7200	171
20.3.3	LIBRERÍA EP93XX_ADC.H	182
20.3.4	LIBRERÍA PEEKPOKE.H	183
<b><u>21</u></b>	<b><u>INDICES</u></b>	<b><u>187</u></b>
<b>21.1</b>	<b>ÍNDICE DE ILUSTRACIONES</b>	<b>189</b>
<b>21.2</b>	<b>ÍNDICE DE TABLAS</b>	<b>191</b>
<b>21.3</b>	<b>ÍNDICE ALFABÉTICO</b>	<b>193</b>
<b><u>22</u></b>	<b><u>LISTADO DE REFERENCIAS</u></b>	<b><u>195</u></b>



# 1 INTRODUCCIÓN

En la actualidad existe una gran preocupación social por el número de personas que mueren anualmente en las carreteras. Para empezar un dato significativo: solo en las carreteras europeas mueren cerca de 40.000 personas cada año, y otras muchas resultan heridas.



**Ilustración 1: Número de muertos en accidente de tráfico en España entre 1992 y el 2008**

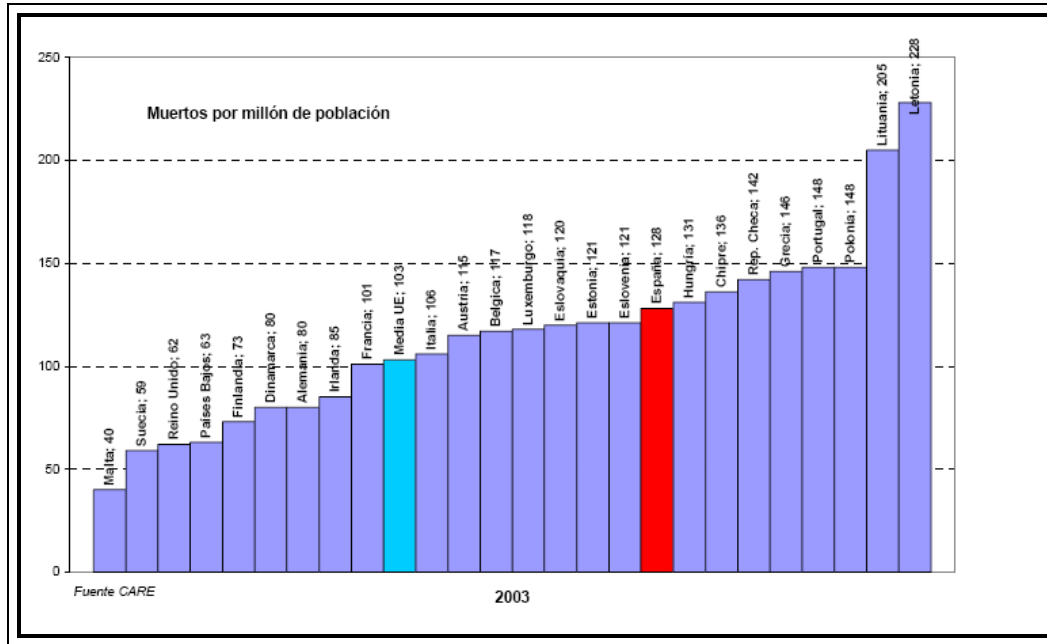
En España mediante la introducción del carnet por puntos, tuvo lugar una notable reducción del número de accidentes. En la *Ilustración 1* puede observarse como desde el año 2000 hasta el 2008 [1] se redujeron prácticamente a la mitad el número de víctimas mortales en accidentes de tráfico<sup>1</sup>, aún así durante el 2008 murieron en accidentes de tráfico más 2000 personas. Puesto que los desplazamientos son inevitables, se hace necesaria la utilización de cualquier medio o dispositivo, que permita reducir, aún más, estas cifras.

## 1.1 POSICIÓN EUROPEA

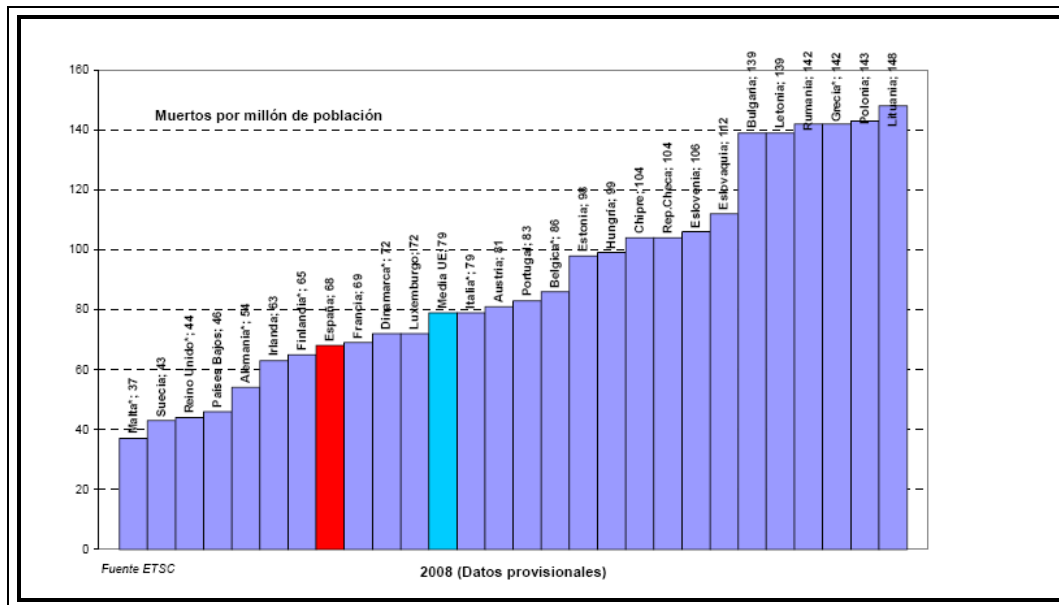
Las muertes en carretera no son sólo un problema en España, sino en toda Europa. Si tenemos en cuenta las muertes por millón de población en el 2003 (*Ilustración 1*) y en el 2008, vemos que se han reducido en toda Europa, pero siguen siendo significativas. En respuesta a la preocupación social que generan estos datos, la comisión europea presentó en el parlamento, en junio de 2008 el informe, *eSafety Forum* [2]. Desde la Comisión Europea se han realizado una serie de acciones con el objetivo de reducir estas cifras. Entre ellas la presentación de la *Carta*

<sup>1</sup> La Dirección General de Tráfico [1] en estos datos, sólo tiene en cuenta los **accidentes mortales** que tienen lugar en la **carretera**, de manera que no se incluyen los accidentes en zonas urbanas, ni aquellos que tienen como resultado exclusivamente heridos, considerándose **víctima mortal** aquella que fallece dentro de las **24 horas** siguientes al accidente.

**Europea de Seguridad Vial.** Donde desde se invita a los países miembros a tomar acciones concretas, que conlleven a reducir la mortalidad en los accidentes de tráfico y a evaluar los resultados de las mismas. En Diciembre del 2008 la comisión presentó el plan de acción para el despliegue de sistemas de transporte inteligentes (ITS, *Intelligent Transportation Systems*) cuyos principales objetivos eran lograr desplazamientos más limpios, eficientes (también en relación a la eficiencia energética) y más seguros. Además, propone un mejor uso de los sistemas de seguridad activa y de asistencia a la conducción, cuyos beneficios están acreditados en términos de seguridad a bordo para los ocupantes del vehículo y de los otros usuarios en carretera.



**Ilustración 2: Muertos por millón de población en el 2003**



**Ilustración 3: Muertos por millón de población en el 2008**

El informe *eSafety Forum*, [2] representa una plataforma, cuya misión es promover el desarrollo de vehículos con sistemas de seguridad inteligentes y que plantea el importante papel que tales vehículos pueden desarrollar en materia de seguridad. Se estima que los vehículos inteligentes podrían salvar hasta 6.500 [3] vidas cada año, en Europa. Las principales cuestiones presentadas en el informe realizado por la comisión sobre vehículos inteligentes fueron: el

desarrollo de un plan europeo de llamadas de emergencia eCall [4], así como reducir el impacto en el medio ambiente, como por ejemplo, en la reducción de emisiones de CO<sub>2</sub>. Además dicho informe insta a las partes involucradas a mejorar la tecnología de los sistemas y a estimular a la demanda de los usuarios, así como la asequibilidad de los mismos.

Existe un gran potencial de mejora de la seguridad vial, mediante el uso de los *sistemas de asistencia a la conducción* [5], tales como: el *control electrónico de estabilidad* (ESC), el *control de la velocidad de cruce* (ACC), los *sistemas de advertencia de colisión y frenado de emergencia*, la *asistencia lateral* (asistente en el cambio de carril y sistema de aviso de cambio involuntario de carril), el servicio de emergencia (eCall) y de los sistemas para combatir la hipovigilancia entre otros.

La tecnología proporciona nuevas soluciones que pueden ayudar a paliar el problema de la alta siniestralidad. El uso de los sistemas activos de seguridad en los vehículos inteligentes tiene como finalidad asistir al conductor para realizar una conducción segura, manteniéndole informado de las posibles situaciones de riesgo.

## 1.2 VEHÍCULOS INTELIGENTES

### 1.2.1 Características de los vehículos inteligentes

Las principales tecnologías que comprenden los “vehículos inteligentes” [6] son:

- ✓ Sistema anti bloqueo de frenos (ABS)
- ✓ Control de cruce adaptativo (ACC)
- ✓ Luces autoadaptables
- ✓ Asistente de cambio de carril. Detección del ángulo muerto
- ✓ Sistemas de control y alerta de somnolencia
- ✓ Sistema de llamada de emergencia del vehículo (eCall)
- ✓ Sistema electrónico de asistencia a la frenada (EBS)
- ✓ Control electrónico de estabilidad (ESC)
- ✓ Sistema de aviso de cambio involuntario de carril (LDWS)
- ✓ Visión nocturna
- ✓ Alerta de velocidad excesiva
- ✓ Sistema de control de la presión en los neumáticos (TPMS)

### 1.2.2 Sistema anti bloqueo de frenos (ABS)

El ABS (Anti-lock Braking System) evita que los neumáticos pierdan adherencia al suelo en el proceso de frenado [7]. Para ello modula automáticamente la presión ejercida sobre los frenos, cuando el conductor pisa el pedal del freno a fondo. Cuando la fricción entre los neumáticos y la superficie de la carretera es baja, las ruedas se bloquean con facilidad y el conductor pierde el control del vehículo. El sistema es capaz de detectar desaceleraciones bruscas del giro de cada rueda y reducir la presión en ese freno hasta que vuelva a detectar una aceleración. Dotando al sistema de la máxima potencia de frenado y reduciendo considerablemente la distancia de frenado. En el año 2006 el 90% de los vehículos europeos llevaban este sistema incorporado [8].

### 1.2.3 Control de crucero adaptativo (ACC<sup>2</sup>)

Esta tecnología ajusta automáticamente la distancia y la velocidad en función de la marcha del vehículo que le precede mediante un detector de radar de gran alcance. Si el vehículo que va delante reduce la velocidad o se detecta otro objeto, el sistema ajusta la velocidad del vehículo incluso llegando a detenerlo, sin necesidad de que el conductor intervenga. Cuando el obstáculo desaparece el vehículo recupera la velocidad programada y se adapta perfectamente al tráfico de nuevo. Avisando, generalmente, mediante una señal acústica. Este sistema puede ser desactivado por el conductor en todo momento [9].

### 1.2.4 Luces autoadaptables

Cuando un vehículo sufre una fuerte desaceleración, el haz de luz de sus faros tiende a bajar, mientras que si acelera bruscamente sube, así mismo cuando toma una curva se desplaza hacia un lado. Las luces autoadaptables orientan el haz luminoso modificando el ángulo de orientación en función de la dirección de las ruedas, el movimiento del vehículo y la velocidad. Con ello se consigue una correcta iluminación de la calzada independientemente del movimiento del vehículo. En la actualidad este dispositivo es opcional en varios modelos europeos [10].

### 1.2.5 Asistente de cambio de carril.

Los sensores, situados en el parachoques de la parte delantera, vigilan que el vehículo no pase de forma involuntaria por encima de las líneas de la carretera como aparece en Ilustración 4, y en caso de hacerlo se activa una señal sonora y una vibración en el asiento que alertan al conductor [29].

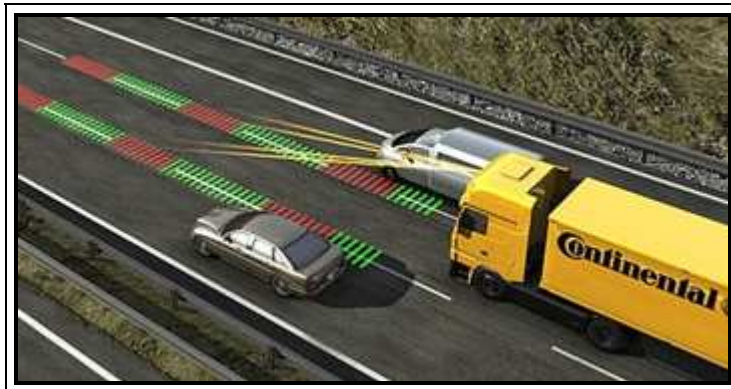


Ilustración 4: Ejemplo ilustrativo del asistente de cambio de carril

### 1.2.6 Detección del ángulo muerto

El sistema vigila los puntos sin visibilidad a ambos lados de la parte posterior del vehículo. Si detecta algún objeto en estos puntos muertos, cuando el conductor ponga el intermitente para realizar un cambio de carril, el volante vibrará ligeramente indicando una situación de potencial peligro [11].

---

<sup>2</sup> ACC corresponde a las siglas en inglés de *Adaptive Cruise Control*.

### 1.2.7 Sistemas de control y alerta de somnolencia

Los sistemas de control y alerta de somnolencia son capaces de detectar una situación de fatiga: analizando el parpadeo, controlando la expresión facial, la posición de la cabeza, el movimiento de manos y pies. La información es analizada y como resultado, el conductor es avisado en caso de que existiera una distracción o un estado de somnolencia [12].

### 1.2.8 Sistema de llamada de emergencia del vehículo (eCall)

Es un sistema diseñado con el fin de reducir el tiempo de respuesta, en caso de accidente, por parte de los sistemas sanitarios. El sistema establece una conexión de voz con el número de emergencias<sup>3</sup>, enviándole los datos del vehículo accidentado y la hora del accidente. Puede ser activado de forma automática o manualmente por los ocupantes del vehículo. A partir del 2015, el sistema eCall deberá ser incorporado en todos los vehículos de venta en la UE [13] [14].

### 1.2.9 Sistema electrónico de asistencia a la frenada (EBS, *Electronic Brake assist System*)

Cuando el conductor quiere detener rápidamente el vehículo, frecuentemente pisa el freno con rapidez pero no con la máxima presión, aumentando la distancia de frenado. Para evitar esta situación, cuando el sistema detecta una frenada brusca aplica la máxima potencia de frenado, e incluso con el simple hecho de detectar una presión moderada disminuirá notablemente la velocidad [15].

### 1.2.10 Control electrónico de estabilidad (ESC<sup>4</sup>, *Electronic Brake assist System*)

En ocasiones el conductor puede perder el control del vehículo, debido a maniobras bruscas para evitar un peligro o a una valoración incorrecta de la peligrosidad de una situación etc. En esas situaciones los vehículos suelen derrapar porque se sobrepasa el límite de tracción de los neumáticos, a veces las ruedas pueden no girar a velocidad suficiente o hacerlo a una excesiva.

El control electrónico de estabilidad detecta la desviación de la trayectoria del vehículo con respecto a la dirección que se deseaba llevar. Es un sistema que depende de la información que le proporciona el ABS [17]. Como resultado de esta interpretación y sin intervención del conductor, actúa sobre cada rueda de forma independiente como se detalla en Ilustración 5, manteniendo la trazada correcta. A menudo el conductor no se da cuenta dicha intervención [18].

---

<sup>3</sup> El número de emergencias corresponde al 112.

<sup>4</sup> ESC corresponde a las siglas en inglés de *Electronic Stability Control*. Fue desarrollado por Bosch y se comercializó con el nombre de *Elektronisches Stabilitätsprogramm*, de ahí ESP, aunque en la actualidad recibe un nombre distinto en función del fabricante del coche en el que se monte : ESC, VDC, VSC y DSC.[16]



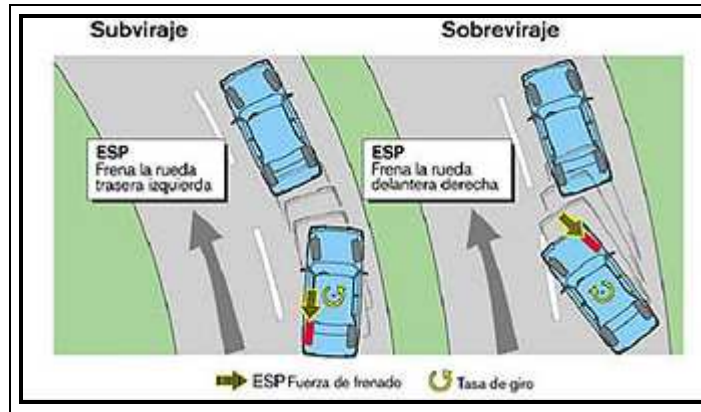


Ilustración 5: Funcionamiento del ESC de Bosch (ESP)

### 1.2.11 Sistema de aviso de cambio involuntario de carril (LDWS)

Estos sistemas vigilan la posición del vehículo en el carril avisando, al conductor si se produce un cambio de carril sin que se haya activado el intermitente. Los LDWS que en la actualidad están comercializados, son sistemas que usan algoritmos para interpretar imágenes de vídeo calculando la posición del vehículo y su alineación con la calzada [19].

### 1.2.12 Visión nocturna

De noche la visibilidad de un conductor se reduce en aproximadamente 40 metros cuando lleva encendidas las luces de cruce. La percepción del color y la profundidad en esta situación disminuye notablemente. Esto puede llevar a detectar un peligro demasiado tarde, sin que al conductor le de tiempo a reaccionar. Por otra parte las luces largas deslumbran a los coches que circulan en sentido contrario, así, su utilización es limitada. Un sistema de visión nocturna introduce una serie de sensores infrarrojos de alcance similar al de las luces largas y no visible para el ojo humano. Una pantalla especial ofrece una mejor visión de la carretera y los posibles obstáculos y puede ayudar a reducir el considerable número de accidentes de carretera que se producen por la noche [20]. Un ejemplo de sistemas de visión nocturna actualmente comercializado es el “BMW Night Vision” que aparece en la Ilustración 6. Permite distinguir objetos que emitan calor, en condiciones favorables, a una distancia de entre 15 y 100 metros [21] [22].



Ilustración 6: Pantalla de BMW Night Vision



### **1.2.13      Sistemas de protección de peatones y usuarios vulnerables en la vía pública**

Los ciclistas y peatones son los usuarios de las vías más vulnerables. Estos sistemas incorporan en la parte delantera del vehículo, varios sensores con el objetivo de detectar los usuarios vulnerables y distinguirlos de otros obstáculos. La protección se realiza mediante diferentes actuadores, como el freno autónomo o el asistente a la frenada para reducir notablemente la velocidad del vehículo antes del impacto. En el caso de que la colisión sea inminente se activan los *accionadores de protección estructural* (airbags en el parachoques, elevación del capó...) con la intención de limitar el daño en estos usuarios. Un ejemplo de desarrollo de estos sistemas es el proyecto AKTIV de Volkswagen que incluye entre otros el desarrollo de un parachoques “blando” y el subproyecto “Pedestrian and Cyclist Safety” que busca un incremento en la seguridad de peatones y ciclistas [23]. En la actualidad la mayoría de estos sistemas no se han comercializado [24].

### **1.2.14      Alerta de velocidad excesiva**

La velocidad es el factor clave en la mayoría de los accidentes de tráfico. El sistema como su propio nombre indica, alerta mediante una señal acústica, visual o táctil (como una vibración) avisando al conductor en el caso en el que se exceda el límite de velocidad legalmente establecido [25].

## **1.3 ¿EXISTE UNA NECESIDAD REAL DE ESTOS SISTEMAS?**

Existen multitud de sistemas inteligentes que aún están en vías de desarrollo, otros no se han comercializado y los que lo hacen están dirigidos principalmente al sector de vehículos de lujo, que representa una pequeña cuota de mercado.

### **1.3.1 Problemas de implantación**

Los principales problemas de implantación son [26]:

- Impedimentos legales.
- Elevados costes de implantación.
- La elevada competitividad del mercado automovilístico en la reducción de costes y por tanto en la reducción del precio final.
- El desconocimiento de los sistemas inteligentes por parte de los usuarios y por tanto su baja demanda de los sistemas.

### 1.3.2 Ventajas de implantación

Por otra parte las principales ventajas de los sistemas inteligentes son [26]:

- Ventajas medioambientales:
  - Reducción del consumo de combustible.
  - Optimización de los desplazamientos.
  - Mejora de los problemas derivados de la congestión de tráfico.
- Ventajas en materia de seguridad vial [26]:
  - Reducir la inseguridad vial
  - Prevención de accidentes y de sus consecuencias
  - Información sobre la red vial en tiempo real

### 1.3.3 Potencial de los sistemas inteligentes

Podrían evitarse más de 1.500 accidentes anuales tan sólo con que el 0.6% de los vehículos incorporase el sistema de aviso de cambio involuntario de carril [27]. Ya que, el 93% de los accidentes se deben a un error humano según estudios recientes, lo que supone un coste aproximado de 200.000 millones de euros. La congestión en las carreteras europeas, supone un gasto de 50.000 millones de euros al año que equivale al 2% del PIB de la UE. Mientras que se calcula, que ciertos hábitos en la conducción y la introducción de nuevas tecnología en los vehículos podría suponer hasta un 50% de reducción en el consumo de combustible [28].

Tanto los vehículos inteligentes como la implantación de la infraestructura necesaria en la carretera pueden aportar importantes beneficios sociales y económicos. Tan solo con la instalación del ACC (sistema de control de cruce adaptativo que ayuda a mantener la distancia con el vehículo que le precede) se prevé que en la unión europea se podrían evitar 4.000 accidentes en tres años, tan sólo con que, estuvieran equipados con este sistema el 3% de los conductores [28].

El sistema de control y alerta de somnolencia se estima que podría evitar el 30% de los accidentes mortales en autopista, esto representaría el 9% de todos los accidentes con víctimas mortales [28].

## 2 INTELLIGENT VEHICLE BASED ON VISUAL INFORMATION

### 2.1 VEHÍCULO IVVI

Las siglas IVVI corresponden al acrónimo en inglés de Vehículo Inteligente basado en información visual. El IVVI es un proyecto desarrollado por la Universidad Carlos III de Madrid, como plataforma de experimentación en el que se han instalado los sensores y equipos necesarios para el desarrollo de Sistemas de Ayuda a la Conducción. En la actualidad se desarrolla la versión 2.0 cuyo logotipo aparece en la Ilustración 7 [32].



**Ilustración 7: Logotipo**

El vehículo, que aparece en la Ilustración 8, está dotado de sistemas basados en el análisis computacional de las imágenes que captan un conjunto de cámaras, incorporadas al vehículo con el objeto de facilitar la ayuda a la conducción.

### 2.2 SISTEMAS QUE INCORPORA EL IVVI

#### 1. Sistema de posicionamiento.

En la mayoría de los casos los sistemas ADAS<sup>5</sup> [31] necesitan o requieren conocer cierta información del vehículo, como puede ser la posición y velocidad en cada instante. Con esta finalidad el vehículo IVVI dispone de un GPS diferencial y un sensor inercial.



**Ilustración 8: IVVI 2.0**



**Ilustración 9: Retrovisor**

#### 2. Sistema de percepción.

Mediante seis cámaras el sistema controlará todo aquello que suceda tanto dentro como fuera del vehículo. Las cámaras son de barrido progresivo, con ello se consigue poder capturar imágenes en movimiento evitando los problemas propios del entrelazado del vídeo.

<sup>5</sup> Los sistemas ADAS son sistemas automáticos de ayuda a la conducción como pueden ser: el detector de ángulo muerto, el avisador del cambio de carril, el control del crucero adaptativo, etc.



**Ilustración 10: Salpicadero**

En la parte delantera del vehículo se sitúa una cámara color cuya función será detectar y analizar la señalización vertical en la carretera y un sistema estereo [30] tanto para la detección de obstáculos como para determinar los límites de la calzada. El sistema dispone de una cámara de infrarrojo lejano que capta el calor que desprenden los objetos, con ella se consigue percibir obstáculos en condiciones adversas de visibilidad.

Además se dispone de una cámara en dirección al conductor tal y como aparece en Ilustración 9. Esta cámara evalúa en todo momento el grado de atención del conductor. La información proveniente de los Sistemas de Ayuda a la Conducción se muestra a través de una pantalla integrada en el salpicadero, como muestra la Ilustración 10.

El GPS proporciona la posición y velocidad a la que circula el vehículo que junto con las imágenes proporcionadas por las cámaras es enviada a los computadores de a bordo, donde los datos son analizados. Proporcionando una percepción del entorno de forma continua y así poder reaccionar en todo momento ante posibles peligros potenciales.

### 3. Sistemas de procesamiento.

Este sistema tiene como objetivo recepcionar las imágenes y procesarlas mediante un conjunto de ordenadores conectados a la red, para una conducción segura. Para ello incorpora en el maletero del vehículo cuatro PCs, tres de ellos situados en el maletero como aparecen en la Ilustración 11.

### 4. Sistema interfaz con el conductor.

La información obtenida por el sistema de procesamiento es suministrada al conductor, aconsejándole sobre la conveniencia o no de la maniobra que está realizando.



**Ilustración 11: Maletero IVVI 2.0**

La alimentación eléctrica de los diferentes equipos y sensores incorporados al vehículo se realiza, mediante una batería adicional incorporada debajo del asiento del copiloto. La batería propia del vehículo se utiliza para alimentar la tarjeta TS-7200 [33], que es la placa controladora del sistema de alimentación que gestiona la información procedente de la batería adicional y de la red de eléctrica. La tarjeta TS-7200 está conectada con los equipos, mostrando al usuario la información relativa a la gestión del sistema de alimentación de los equipos mediante una interfaz gráfica. El sistema de alimentación incluye un inversor que transforma 12V en continua a 220V en alterna para la alimentación de los equipos que se alimenten a dicha tensión. Si la red de alimentación no proporcionase tensión actuaría una SAI que mantendría alimentados ininterrumpidamente a los equipos críticos en caso de que existiera una falta en el suministro de energía eléctrica.

## 3 SISTEMA DE GESTIÓN DE BATERÍA

### 3.1 DISEÑO

Para realizar un diseño eficaz del sistema de alimentación se han realizado los siguientes pasos que aparecen detallados en la Ilustración 12.

#### 3.1.1 La idea

En primer lugar partimos de la idea: La necesidad de alimentar unos equipos que se han introducido en el vehículo.

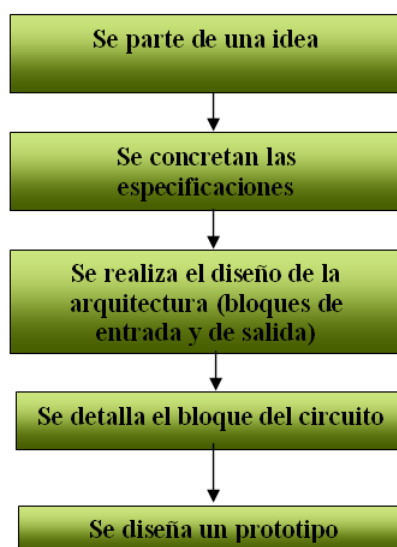


Ilustración 12: Proceso para el diseño eficaz

#### 3.1.2 Las especificaciones

Como ya se ha señalado con anterioridad, se han incorporado al vehículo sensores y equipos extras que conforman el Sistema de Ayuda a la Conducción. Los equipos que se han añadido, no deberían alimentarse con la batería propia del vehículo para no comprometer la capacidad de arranque del mismo. Por ello, se ha incorporado una batería adicional debajo del asiento del copiloto. Algunos de los equipos tienen un funcionamiento crítico, por lo que se hace necesario incluir un sistema de alimentación ininterrumpida, también conocido como SAI, para mantener el suministro eléctrico en estos equipos, en el caso en el que falle dicho suministro. Por otro lado, se necesitan 220V de tensión alterna para alimentar a los equipos extras y puesto que la batería adicional proporciona una tensión continua, se incorpora un inversor al sistema que permita la conversión de tensión continua a tensión alterna.

El sistema SAI se alimentará según las necesidades del usuario directamente de la red de suministro o a través del inversor, que a su vez está conectado a la batería situada debajo del asiento del copiloto.

Una placa controladora (el modelo TS-7200 [33]) gestionará el sistema de alimentación de la batería adicional, mediante un interfaz usuario que se comunica con los equipos a través del protocolo TCP/IP.

Estas especificaciones básicas nos llevan a desarrollar la arquitectura del diseño que aparece en la Ilustración 13.

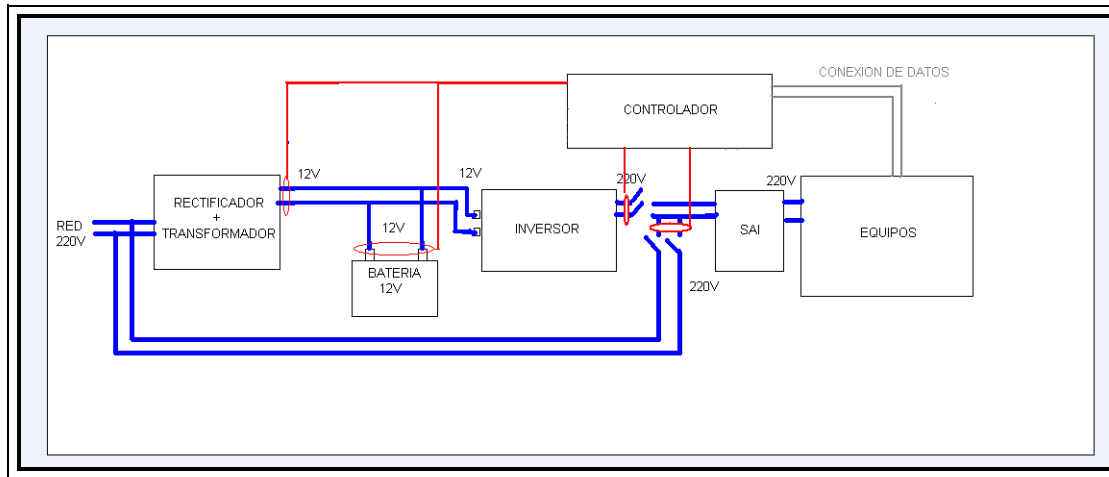


Ilustración 13: Arquitectura del diseño

### 3.1.3 Arquitectura del diseño

La energía eléctrica que se necesita para el funcionamiento de los sistemas puede provenir de dos fuentes diferentes en función de si el vehículo se encuentra en el laboratorio o circulado por la calle. En el primer caso, la energía eléctrica se obtendrá de la red de 220V del laboratorio y en el segundo caso la energía eléctrica se obtendría de la batería situada debajo del asiento del copiloto. Dado que ambas fuentes de energía son de un cariz muy diferente el flujo de la corriente eléctrica será muy diferente (ver Ilustración 13).

Cuando el vehículo se encuentre en el laboratorio estará conectado a la red y la corriente eléctrica entrará directamente en la SAI que sirve para proteger a los equipos ante sobretensiones o posibles caídas de la red. Todos los equipos críticos se encuentran conectados a la SAI.

Si el vehículo se encuentra circulando por la calle, la corriente que suministra la batería adicional es continua a 12V, por lo que, en primer lugar se debe transformar a 220V alterna, esto se hace mediante un inversor. Una vez que se dispone del voltaje y de la forma de onda adecuada la tensión llega a la SAI y finalmente a los equipos.

Como la batería necesita ser recargada cada cierto tiempo, también se dispone de un cargador de baterías, de tal manera que cuando el vehículo se encuentra conectado a la red del laboratorio pueda recargarse. Dado todos los posibles modos de funcionamiento es necesario implementar un sistema de control que detecte en cada momento el modo de funcionamiento y gestione el flujo de energía sin que se produzcan cortocircuitos por una conexión indebida.

## 4 BATERIA

### 4.1 INTRODUCCIÓN

Las baterías de plomo ácido para automóvil, también llamadas acumuladores, son dispositivos que almacenan energía eléctrica mediante procedimientos electroquímicos. Este tipo de baterías son las más utilizadas en automoción debido a su bajo coste.

### 4.2 ELEMENTOS DE UNA BATERÍA COMERCIAL DE PLOMO-ÁCIDO

La mayor parte de las baterías comerciales de Plomo-Ácido están formadas por seis elementos separados entre sí por celdas (ver Ilustración 14) [34]. Cada elemento está compuesto por varias placas de plomo positivas y negativas separadas por un material plástico aislante. Los grupos de placas y los elementos están unidos mediante conectores. Las placas separadoras evitan cortocircuitos que aceleran la descarga de la batería. Cada elemento excede levemente de 2V dependiendo de su carga.

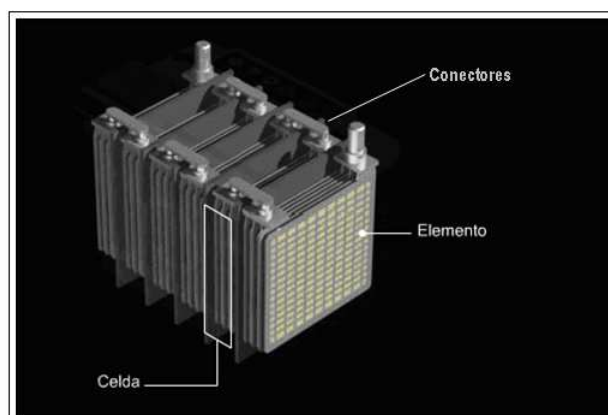


Ilustración 14: Componentes de una grupo

Las placas positivas se alternan con las negativas. Los conectores unen los elementos formando un circuito serie de dos polos que termina en los bornes o terminales uno positivo y otro negativo claramente identificados (ver Ilustración 15). La suma de las tensiones son 12 voltios.

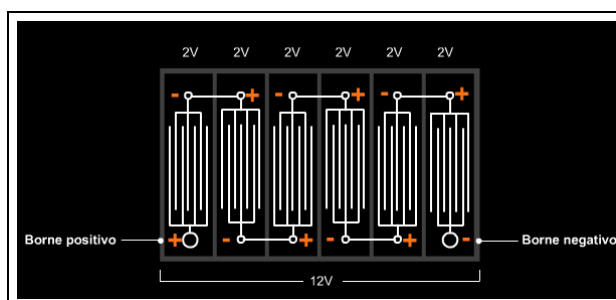


Ilustración 15: Detalle de funcionamiento de una batería



Los elementos se encuentran sumergidos en una solución electrolítica. La carcasa se compone de unas formas sinusoidales (ver Ilustración 16) que conforman la cámara de decantación, donde los residuos de la reacción química se depositan en el fondo, de esta forma se evitan que entren en contacto con las placas y den lugar a cortocircuitos.

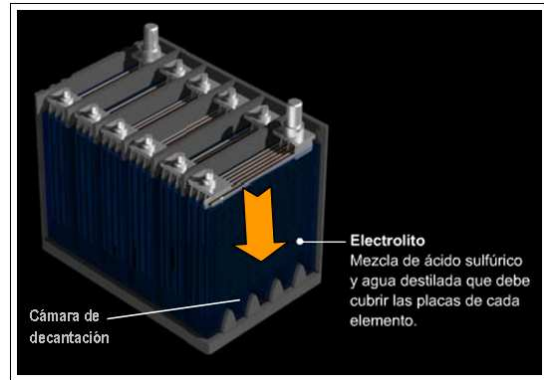


Ilustración 16: Detalle de los componentes de una batería

Entre los acumuladores de plomo-ácido se pueden distinguir:

- Baterías de arranque de automóviles.
- Baterías de tracción en vehículos eléctricos.
- Baterías estacionarias utilizadas en sistemas de alimentación ininterrumpida.

### 4.3 FUNCIONAMIENTO DE UNA BATERÍA COMERCIAL DE PLOMO-ÁCIDO

En las baterías de plomo ácido los dos electrodos sufren un proceso de reducción y oxidación. El cátodo (negativo) compuesto de dióxido de plomo y el ánodo (+) de plomo metálico se reducen y se oxidan respectivamente en el seno de un electrolito de ácido sulfúrico transformándose en sulfato de plomo y agua.

Cuando la batería está cargada el electrodo positivo tiene un depósito de dióxido de plomo y el negativo es plomo. Al descargarse la reacción química que tiene lugar hace que tanto en el electrodo positivo como en el negativo tengan un depósito de sulfato de plomo. La Ilustración 17 muestra estos dos estados. El tapón de ventilación es necesario ya que en el proceso químico libera gases que es necesario llevar al exterior de la batería.

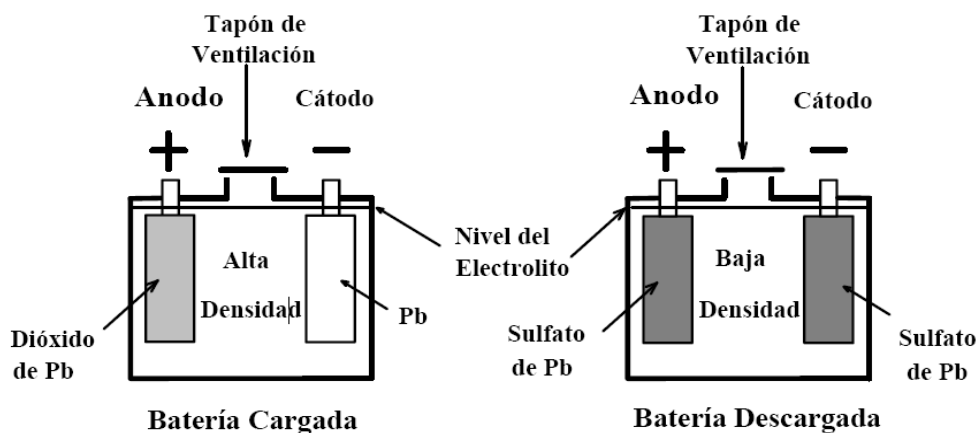


Ilustración 17: Estados de una batería



A medida que se consume el ácido disminuye la densidad, tal y como se muestra en la Ilustración 17. La tensión, la capacidad y la corriente máxima que puede proporcionar una batería disminuyen conforme se va formando el sulfato de plomo.

### 4.3.1 Proceso de carga y descarga

El proceso de carga se realiza mediante un generador de corriente continua conectado a la batería. Para crear una *corriente de carga* el voltaje deberá ser ligeramente superior al de la batería. La *corriente de carga* provoca reacciones químicas en los electrodos hasta que el electrolito es incapaz de mantener las reacciones o cesa la corriente. El proceso es reversible. Al desconectar el generador y conectar una carga a la batería circula una corriente a través de la carga, en dirección opuesta a la dirección de carga devolviendo el proceso a la situación inicial [112]. El *ciclo de carga y descarga* puede ser repetido durante un número limitado de veces ya que con cada descarga los electrodos pierden material [35]. Es conveniente cargar una batería con un nivel de corriente que no exceda el máximo aconsejado por el fabricante.

### 4.3.2 Influencia de la temperatura en la batería

La temperatura de trabajo supone un problema tanto si está por encima de 30 °C como por debajo de 0°. Cuando la batería está descargada disminuye la cantidad de ácido en la solución, por tanto la cantidad de agua en el electrolito será mayor. Al bajar la temperatura existe la posibilidad de que se congele, si esto ocurre el volumen aumenta y la fuerza de expansión distorsiona los electrodos pudiendo dañar los bornes. Una batería de plomo ácido totalmente descargada se congela alrededor de los -10°C y si esta completamente cargada alrededor de los -58°C (ver Tabla 2) [35]. Por encima de 25°C se produce una mayor actividad del electrolito<sup>6</sup> lo que se traduce en una reducción de la vida útil. De acuerdo con la Tabla 1 a 30°C se reduce un 30% la vida útil de la batería. Entendiendo como vida útil, el tiempo total que durará la batería antes de tener que sustituirla. La vida útil de una batería se basa en el número de ciclos de carga y descarga que es capaz de completar manteniendo más del 80% de su capacidad original [36].

BATERÍA PLOMO ÁCIDO	
Temperatura del electrolito (°C)	Reducción de la vida útil (%)
25	0
30	30
35	50
40	65
45	77
50	87
55	95

Tabla 1: Características de una batería de plomo ácido. Temperatura y vida útil

<sup>6</sup> Se debe recordar que en cualquier reacción química el incremento de la temperatura provoca una aceleración de la reacción así mismo, una disminución de la temperatura da lugar a una desaceleración de la reacción.

BATERÍA PLOMO ÁCIDO	
Temperatura de congelamiento del electrolito (°C)	Estado de carga (%)
-58	100
-34,4	75
-20	50
-15	25
-10	Descargada

Tabla 2: Características de una batería de plomo ácido. Temperatura de congelación y estado de carga

La temperatura del electrolito en una batería también está relacionada con el rendimiento, como muestra la Tabla 3, a 26.6 °C el rendimiento es máximo. A medida que la temperatura disminuye lo hace también el rendimiento, prácticamente linealmente como se apreciar en la Ilustración 18 [37]

BATERÍA PLOMO ÁCIDO	
Temperatura del electrolito (°C)	Rendimiento de la batería (%)
26.6	100
0.0	65
-17.7	40
-28.8	18

Tabla 3: Características de una batería de plomo ácido. Temperatura y rendimiento

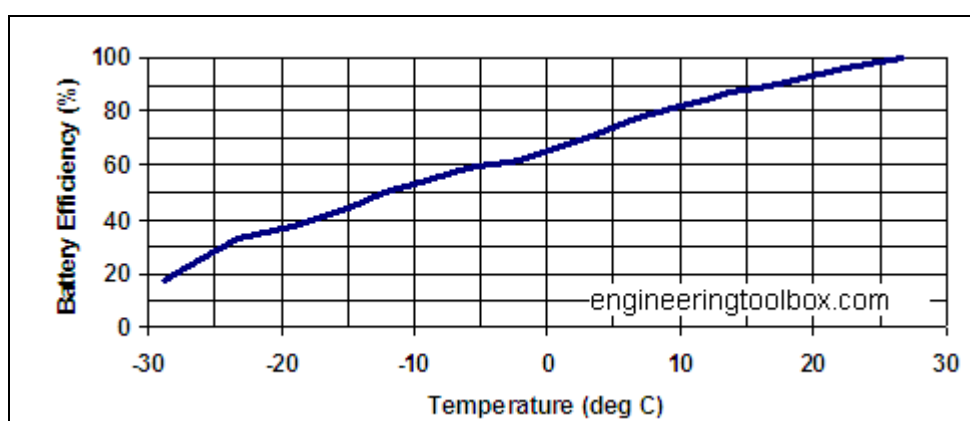


Ilustración 18: Gráfica de rendimiento de una batería de plomo ácido con respecto a la temperatura en °C

## 4.4 LA CAPACIDAD EN LA BATERÍA

Las baterías se caracterizan por su capacidad. La capacidad es una medida de la cantidad de energía eléctrica que es capaz de almacenar una batería. La unidad de carga eléctrica en el

Sistema Internacional de Unidades es el Culombio (C), su valor es muy pequeño y por tanto las baterías suelen emplear como unidad el amperio-hora (Ah) que equivale a 3600C.

#### 4.4.1 Tiempo de duración de una batería

El tiempo de operación de una batería depende de su capacidad. Para calcularlo son necesarios conocer, la capacidad de la batería (Ah), y el consumo eléctrico de los dispositivos (A).

$$\text{Tiempo(horas)} = \frac{C_{\text{batería}} (A \cdot h)}{\text{Consumo}(A)}$$

La corriente de carga y de descarga de una celda se describe frecuentemente como un múltiplo de la **capacidad nominal**  $C_n$  de la batería. Los fabricantes proporcionan el dato de la capacidad nominal para cargas y descargas de distinta duración mediante la letra C seguida de un número, que representa la capacidad de la batería para una descarga de ese número de horas. Este tiempo corresponde a un valor de capacidad de la batería del 100%, si el porcentaje de carga varía, el tiempo variará porcentualmente al valor de carga.

Por ejemplo:  $C_{100}$  representa la capacidad de la batería para una descarga de 100 horas. Así mismo la corriente  $0,1 C_n$  para una celda con capacidad nominal de 1,4 Ah es de 140mA.

#### 4.4.2 Variación de la capacidad de la batería con la temperatura

La capacidad de una batería depende tanto de la temperatura como del régimen de carga y descarga. Si la capacidad disminuye, la descarga se hará más rápidamente. La capacidad se verá afectada también, por el número de ciclos acumulados de carga y descarga. En la Ilustración 20, las curvas de descarga muestran que a baja temperatura la caída del voltaje es mucho más severa que la que se observa a 25°C (ver Ilustración 19). La baja temperatura retarda la reacción química, que equivale a un aumento de la resistencia interna que da lugar a una mayor caída del voltaje [38]. Si se mantiene la temperatura del electrolito constante la caída del voltaje es mayor cuando la corriente de descarga aumenta.

Para poder conocer el valor de la *corriente de descarga* que aparece en las graficas de la Ilustración 19 y de Ilustración 20 se debe seleccionar el tiempo de descarga en horas (100, 20, 10, 5 y 3 horas) y sustituir C por el valor de la capacidad de la batería. Así, si el tiempo de descarga es de 10 horas (valor suministrado por el fabricante como, *Capacidad nominal*,  $C_n$ ) y la capacidad de la batería es de 90 Ah, la corriente será de 9A y la *curva de corriente de descarga* que le corresponde será aquella que se indique C/10 en las curvas de descarga. Para la lectura de la *corriente de carga* se haría de manera análoga con las *curvas de corriente de carga* y el valor del *tiempo de carga* [38].

Las baterías sufren un efecto de *autodescarga* cuando están cargadas y no están siendo utilizadas. La rapidez de la descarga depende de la temperatura ambiente y del tipo de batería [35]. El final de la batería se establece en el momento en el que su capacidad se ha reducido en un 80% respecto a su capacidad nominal.

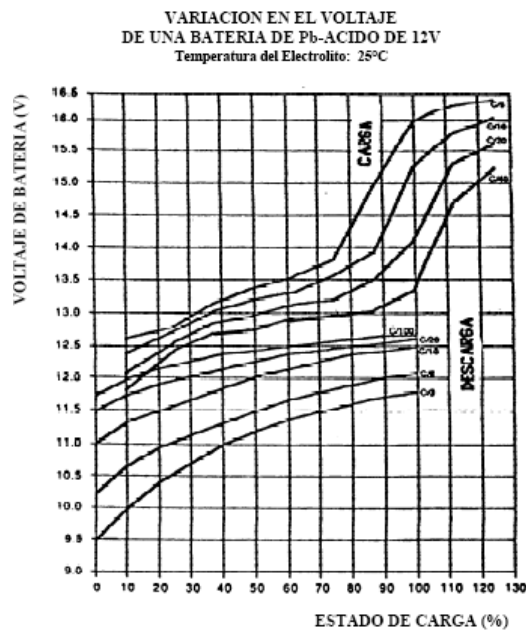


Ilustración 19: Variación del voltaje a 25 °C

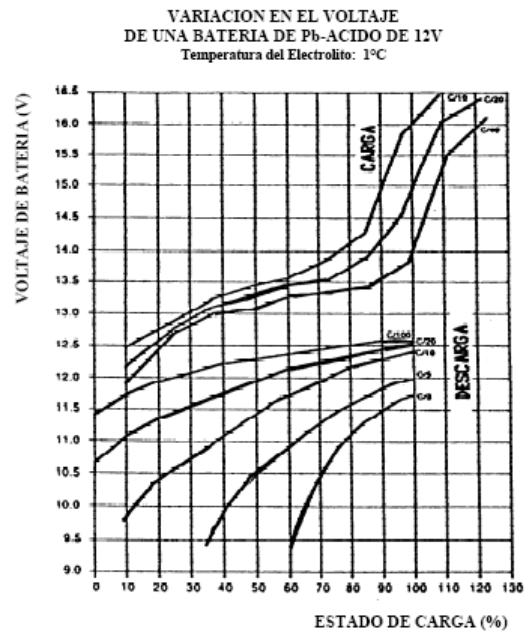


Ilustración 20: Variación del voltaje a 1 °C

#### 4.4.3 Estimación de la capacidad según el voltaje

La capacidad de una batería de vehículo aproximada se puede estimar a partir del voltaje entre los bornes. Este dato es bastante fiable si la batería no tiene una resistencia interna muy alta. La resistencia interna será alta si cae el voltaje rápidamente a corrientes reducidas.

Tensión en bornes	Capacidad aproximada
12.65V	100%
12.45V	75%
12.24V	50%
12.06V	25%
11.89V	0%

Tabla 4: Relación entre la capacidad y tensión en bornes de una batería de vehículo

### 4.5 LA BATERÍA ADICIONAL

Se dispone de una batería de plomo para vehículos de categoría J, una tensión de 12V, una capacidad de 90Ah y una corriente máxima de descarga de 720A. Se utilizará como batería adicional para alimentar los circuitos y no sobrecargar la batería propia del vehículo y se colocará debajo del asiento del copiloto.

## **5 SENSOR DETECTOR DE RED**

### **5.1 DISEÑO DE UN SENSOR DETECTOR DE RED**

#### **5.1.1 Objetivo**

El correcto funcionamiento del sistema de alimentación presenta la necesidad de conocer de antemano si la red de alimentación está proporcionando tensión. El objetivo del diseño es obtener un circuito que detecte si la red de alimentación está proporcionando tensión, esta información deberá acondicionarse para que la tarjeta TS-7200 [33] (bloque controlador) pueda interpretarla y trabajar con ella.

0V → No hay tensión de red.  
3.3V → Si hay tensión de red.

Si la placa TS-7200, está alimentada, cuando le llegue un valor de 3.3V a su entrada digital deberá informar al usuario que el sensor está conectado a la red. Si en cambio recibe un valor muy bajo próximo a cero la tarjeta TS-7200 deberá informar que el sensor no está conectado a la red. Podría ocurrir el sensor estuviera aportando 3.3V y la placa no estuviera alimentada. En tal caso esta información llegaría al usuario a través del interfaz gráfico mediante un panel de aviso (para más información consultar apartado 17.2.1).

### **5.2 DIAGRAMA DE BLOQUES DEL SENSORED**

El diseño del circuito del sensor detector de red se ha basado en el funcionamiento de las fuentes reguladas. La señal de entrada será la red de alimentación que deberá ser acondicionada conforme a la Ilustración 21. El acondicionamiento consiste en la reducción, rectificación y filtrado de la señal de entrada para obtener un valor de tensión que se comparará con la tensión de referencia,  $V_{ref}$ . El comparador necesita una alimentación que obtendrá de la batería del vehículo que se llamará  $V_{bat}$ . Como resultado de la comparación se obtendrá información sobre si el SENSOR de RED se encuentra conectado a la red de alimentación o no.

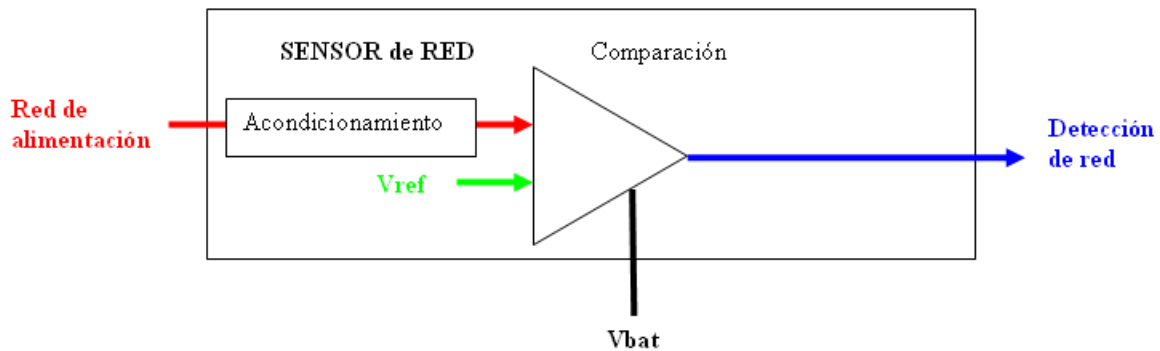


Ilustración 21: Diagrama del sensor de red

### 5.3 RESTRICCIONES QUE CONDICIONAN EL DISEÑO

La tarjeta TS-7200 dispone de una serie de entradas digitales, que permiten una señal de entrada entre 0 y 3.3V [39]. Este dato, condiciona la salida del SENSOR de RED, la cual, deberá estar comprendida en ese mismo rango. Otra restricción es la alimentación del SENSOR de RED. El coche dispone de una salida de 12V en continua, que será la que se utilice tanto en la alimentación de la placa TS-7200 como en la del SENSOR de RED.

En general, a la hora de seleccionar los componentes es necesario tener en cuenta las restricciones que suponen los componentes en stock de los que se dispone o la falta de facilidades en el suministro por parte de determinados proveedores.

Conforme a las restricciones de diseño se busca un circuito de bajo consumo y reducido tamaño. Además, en todo momento se intentará trabajar con tensiones y corrientes que no pongan en riesgo la integridad de aquellos quienes manipulen el circuito.

### 5.4 DESARROLLO

En realidad, detectar si el vehículo está conectado a la red o no, no es más que comparar una entrada de tensión con una tensión de referencia de forma reiterativa. Esta comparación se realiza mediante un amplificador operacional convencional, que se alimenta con valores de tensión continua, en este caso  $V_{bat}$  (tensión continua de valor 12V, procedente de la batería propia del vehículo). La señal de entrada como ya se ha señalado es la tensión procedente de la red de alimentación. La red de alimentación en España es de 230V<sub>p</sub> a 50Hz [71], esta será la entrada al primario del transformador para reducir la tensión a valores más seguros al introducir un aislamiento galvánico y adecuar la tensión al circuito de medida. Además será necesario rectificarla y filtrarla antes de obtener valores de continua admisibles en el amplificador operacional.

## 5.5 ACONDICIONAMIENTO DE LA ENTRADA

### 5.5.1 Diagrama de bloques

El acondicionamiento de la señal de entrada se compone de cuatro bloques fundamentales que aparecen en la Ilustración 22 :

- Bloque de reducción de la señal de entrada.
- Bloque de rectificación.
- Filtro.
- Bloque regulador.

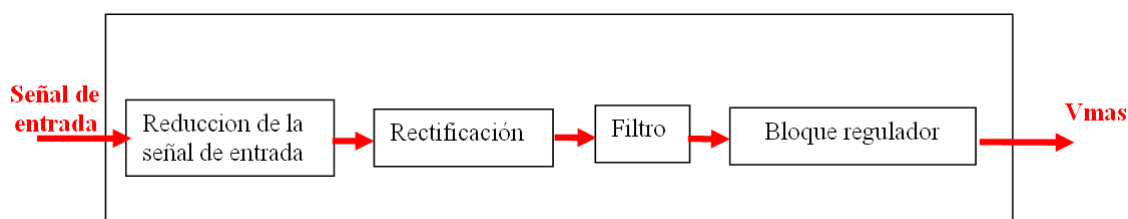


Ilustración 22: Diagrama de bloques del acondicionamiento de la señal de entrada

En la Ilustración 22 la señal de entrada es la señal procedente de la red y la salida  $V_{mas}$  es la señal de continua que entra en el amplificador operacional, ver esquema eléctrico en la Ilustración 25.

## 5.6 BLOQUE DE REDUCCIÓN DE LA SEÑAL DE ENTRADA

La red de alimentación proporciona  $230V_p$  a 50Hz, este valor no es adecuado para trabajar con los elementos de instrumentación, además de ser potencialmente peligrosa su manipulación. Por ello deben reducirse los valores de tensión. La reducción se realiza mediante un transformador aislando galvánicamente las dos partes del circuito. El transformador debe ser de tamaño reducido y precio asequible.

### 5.6.1 Simulación en OrCAD de un transformador

Para simular en OrCAD<sup>7</sup> [40] un transformador de forma aislada, es necesario colocar una resistencia en serie entre el devanado primario ( $R_t$ ) y la fuente de tensión al transformador y otra en paralelo ( $R_L$ ) con el devanado secundario. Para que las resistencias no afecten al circuito se debe cumplir que la impedancia inductiva sea mucho mayor que la impedancia resistiva, de manera que aplicado al circuito primario  $\omega L$  deberá ser mucho mayor que la resistencia  $R_t$  [41]. El valor de  $R_t$  deberá por tanto pequeño (inferior a  $1\Omega$ ) siempre y cuando permita la simulación. La simulación de un transformador ideal utiliza un valor de acoplamiento cercano o igual a 1 ( $K=1$ ). Además el circuito secundario necesita una conexión a tierra DC si se simula de forma aislada.

<sup>7</sup> OrCAD es un software utilizado para crear circuitos electrónicos.

El valor de  $R_L$  no será necesario introducirlo en el momento en que la carga sea introducida. Como ya se ha explicado  $R_L$  debe cumplir la inecuación (1), para que la impedancia de la carga sea mucho menor que la impedancia en el secundario:

$$\omega L_2 \gg R_L \quad (1)$$

Donde  $R_L$  es la resistencia de carga en el secundario en  $\Omega$ ,  $\omega$  es la pulsación en rad/s y  $L_2$  la inductancia en el secundario en H.

El valor de  $R_t$  de forma análoga cumplirá la inecuación (2) siendo  $R_t$  la resistencia interna en el primario expresada en  $\Omega$ ,  $\omega$  la pulsación (rad/s) y  $L_1$  la inductancia en el primario (H).

$$\omega L_1 \gg R_t \quad (2)$$

En OrCAD la relación de transformación no viene dada por la relación de tensiones sino por la relación de inductancias. Para obtener la relación entre tensiones e inductancias se han considerado los siguientes términos:

- $U_1$  es la tensión en el primario del transformador.
- $U_2$  es la tensión en el secundario del transformador.
- $I_1$  es la corriente que circula por el primario.
- $I_2$  es la corriente que circula por el secundario.
- $L_1$  es la inductancia correspondiente al devanado primario.
- $L_2$  es la inductancia correspondiente al devanado secundario.

La relación entre las tensiones de los bobinados y la corriente que circula por los mismos viene dada por la ecuación (3). Además se conoce la relación entre la tensión en cada bobinado con su respectiva inductancia, ver ecuaciones (4) y (5).

$$\frac{U_1}{U_2} = \frac{I_2}{I_1} \quad (3) \qquad U_1 = L_1 \cdot \frac{\Delta I_1}{\Delta t} \quad (4) \qquad U_2 = L_2 \cdot \frac{\Delta I_2}{\Delta t} \quad (5)$$

Relacionando las ecuaciones (3) (4) y (5) obtenemos la inducción en el devanado primario en función de la inducción el devanado secundario y las tensiones en el primario y en el secundario (6):

$$\frac{L_1 \cdot \frac{\Delta I_1}{\Delta t}}{L_2 \cdot \frac{\Delta I_2}{\Delta t}} = \frac{U_1}{U_2} \quad \boxed{L_1 = L_2 \cdot \left( \frac{U_1}{U_2} \right)^2} \quad (6)$$

En OrCAD,  $L_1$  se corresponde con el valor L1\_VALUE y  $L_2$  se corresponde con el valor L2\_VALUE.

*Nota:*

*Se ha considerado que el acoplamiento es 1 ya que en los transformadores convencionales este valor se aproxima a la unidad. En OrCAD se configura el acoplamiento introduciendo en el parámetro COUPLING un uno. Si se deseara introducir un acoplamiento no ideal, se debería introducir un componente denominado*



“K\_linear” que se encuentra en la librería ANALOG, el cuál no debe ir conectado. Mediante el editor de propiedades (botón derecho → Edit properties) se determinan cuales son los conductores que van a acoplarse, para ello se completan las columnas con el nombre del inductor que se desea acoplar (L1, L2 en este caso).

Finalmente, se fija la tensión de salida deseada y se decide el voltaje con el que se va a trabajar sea de  $6V_{p\text{ eff}}$  por ser una tensión manejable. De forma que:

- $U_1$  tensión en el devanado primario corresponde a la tensión que proporciona la red de alimentación:  $230 V_{p\text{ eff}}$
- $U_2$  tensión de salida sea entre  $5-7 V_{p\text{ eff}}$

### 5.6.2 Selección del transformador

En un primer lugar se selecciona un transformador de marca *Myrra* como el que aparece en la Tabla 6. Con el se realiza un primer prototipo. Este modelo finalmente se descarta al delimitar las dimensiones, fijándose unas dimensiones de diseño inferiores a  $30 \times 30 \times 30 \text{ mm}$  y de coste inferior a 5 € por unidad. Se sustituye por un modelo más económico y de reducido tamaño de la marca *saber* [43] [44] cuyo distribuidor Ariston es uno de los proveedores habituales de la Universidad. La Tabla 6 es un cuadro comparativo entre ambos condensadores que refleja las dimensiones, precio unitario y características técnicas de ambos transformadores.

El transformador seleccionado tiene un valor de inductancia en el devanado primario de  $33.3\text{H}$  [42]. De forma que se recalcula el valor predefinido para las inductancias según la ecuación (6). Despejando conforme a la ecuación (7) se obtiene el valor de la inductancia en el secundario.

$$L_1 = L_2 \cdot \left( \frac{U_1}{U_2} \right)^2 \Rightarrow L_2 = L_1 \cdot \frac{U_2^2}{U_1^2} = 33.3 \cdot \frac{6^2}{230^2} = 22.66\text{mH} \quad (7)$$

Para realizar la simulación en OrCAD se utilizan los valores de las inductancias del devanado primario y del secundario, que aparecen en Tabla 5, en el circuito de simulación de la Ilustración 23.

L1_VALUE	33.3H
L2_VALUE	22.66mH

Tabla 5: Valores de inductancias

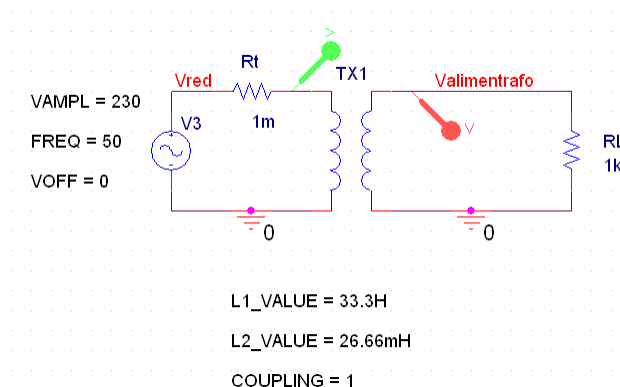


Ilustración 23: Circuito de simulación del transformador

En la Ilustración 24 se pueden observar el resultado de la simulación del transformador aislado (con una carga ficticia de  $1\text{K}\Omega$ ) donde la forma de onda se

conserva, reduciendo el valor de la tensión con relación de transformación de 230 a 6,5V.

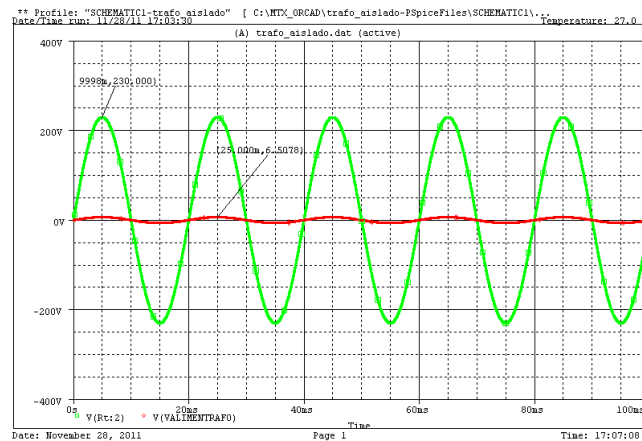


Ilustración 24: Relación de transformación del transformador aislado<sup>8</sup>


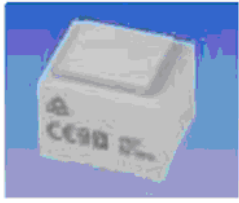
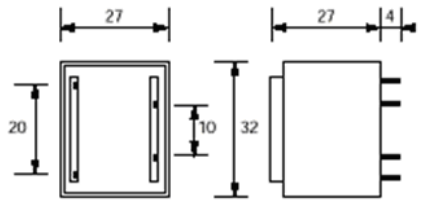
CUADRO COMPARATIVO		
	Transformador myrra	Transformador Saber
Dimensiones (mm)	36.9×32.5×44.1	30×25×15
Precio unitario	7.18€	2.97€
Características técnicas	<p>MYRRA - 44235 - TRANSFORMADOR, 5VA, 230V, 2 X 6V</p>  <p>Fabricante: MYRRA Referencia de fabricante: 44235 Conformidad RoHS: ● Sí</p> <p><b>Descripción</b></p> <ul style="list-style-type: none"> <li>TRANSFORMADOR, 5VA, 230V, 2 X 6V</li> <li>Tipo de transformador: Encapsulado</li> <li>tensión de entrada -: 230V</li> <li>tensión de salida: 6V</li> <li>Rango de corriente: 32mA</li> <li>Potencia nominal: 5VA</li> <li>AC Power: 5VA</li> <li>External Depth: 36.9mm</li> <li>External Length / Height: 32.5mm</li> <li>External Width: 44.1mm</li> <li>Tipo de montaje: PCB</li> <li>Power per Secondary Winding: 2.5VA</li> <li>Secondary Current Rating: 417mA</li> <li>Secondary Voltages: 6V</li> <li>Single Primary Voltage: 230V</li> <li>Longitud: 32.25mm</li> </ul> <p><b>Precio</b></p> <p>Cantidad 1 - 24 Precio unitario 7,18 €</p>	<p><b>TRANSFORMADOR</b> <u>Capsule transformers</u> <b>saber</b></p> <ul style="list-style-type: none"> <li>Circuito impreso miniatura 2,4 VA</li> <li>Primario 220 V</li> <li>Núcleo de 30 x 25 x 15</li> </ul> <p>TR4106 6 V</p> <p>Lote: 1 Ud. 2,9700 €</p>  

Tabla 6: Comparativa de transformadores

<sup>8</sup> Simulación realizada en un tiempo de ejecución de 100ms y un escalón de paso máximo de 1μs.

## 5.7 BLOQUE DE RECTIFICACIÓN

Una vez reducida la tensión, el siguiente paso es transformar la corriente alterna del secundario en una corriente rectificada de onda completa. La rectificación se consigue mediante un puente de diodos con el que se obtiene una rectificación de onda completa, que duplica el ciclo útil ya que tanto los ciclos positivos como negativos proporcionan potencia de salida. El origen de la duplicidad del ciclo útil es que la frecuencia de salida se duplica con respecto a la frecuencia entrada. De acuerdo con la ecuación (8) la frecuencia a la salida de la rectificación serán 100Hz y el periodo 10ms.

$$f_{salida} = 2 \cdot f_{entrada} = 2 \cdot f_{red} = 2 \cdot 50 = 100Hz \Rightarrow T_{salida} = \frac{1}{f_{salida}} = 10ms \quad (8)$$

El puente de diodos seleccionado es el GBPC2506 [80] aunque la simulación se ha realizado con el componente S20WB60 [71] ya por presentar un comportamiento similar al GPC2506. El GBPC2506 se seleccionó por su capacidad para soportar la caída de tensión directa de red realizando un sobredimensionamiento que permite cambiar las características del transformador en futuras aplicaciones. Además este componente estaba dentro del listado de componentes facilitados por los proveedores habituales de la Universidad mientras que el S20WB60 no lo estaba.

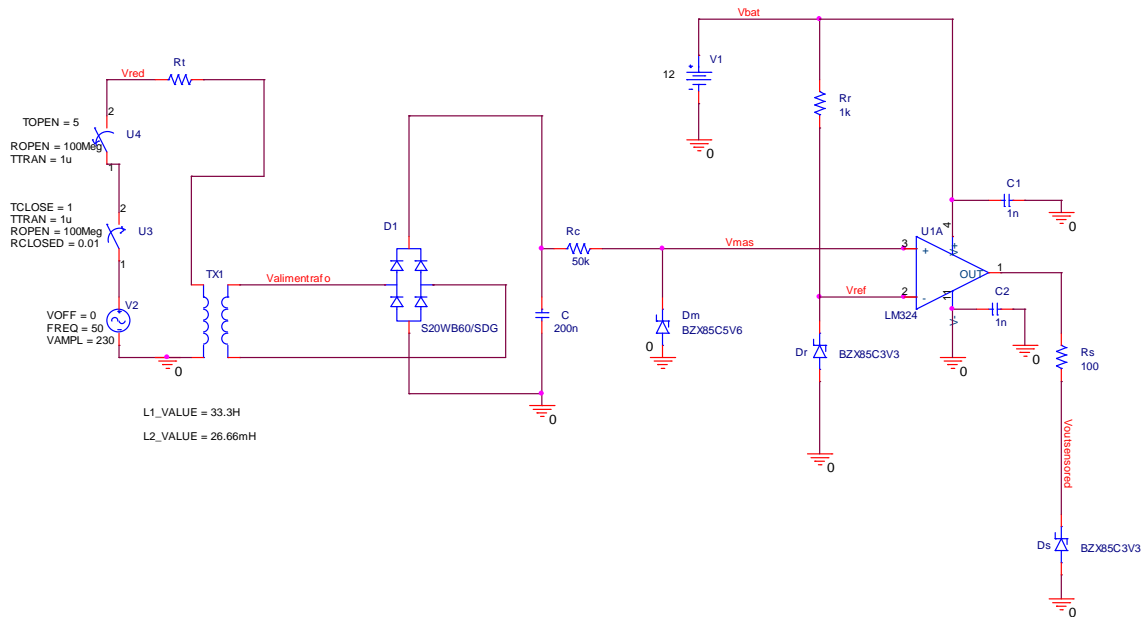
La simulación es posible al ser las condiciones del modelo S20WB60 más restrictivas que el GBPC2506 como se puede ver Tabla 7 donde aparecen los valores máximos de tensión en DC ( $V_{dc}$ ) y los valores máximos de corriente directa ( $I_f$ ) que soporta el componente. Si se desea ser más exactos en los cálculos, se debe situar el cursor sobre el componente y presionar *Edit* en el menú principal. *Edit* → *PSPice Model* y en el modificar el texto de acuerdo con las fórmulas que aparecen en la página 134 (*Diode equations*) basadas en la constante de Boltzman del Manual de Referencia OrCAD PSPice A/D [113].

MODELO	$V_{dc}$	$I_f$
S20WB60	600V	20A
GBPC2506	600V	25A

Tabla 7: Comparación entre los modelos S20WB60 y GBPC2506

## 5.8 BLOQUE DE FILTRADO

A la salida del puente de diodos se sitúa la etapa de filtrado con la intención de eliminar la componente alterna de la señal. Esto se logra colocando un condensador  $C$  en paralelo con la carga de acuerdo con la Ilustración 24. Para simular el comportamiento del sistema durante la detección de la señal de red, se ha simulado el sistema estableciendo que la señal de red aparece un segundo después de que el sensor sea alimentado y permanece conectado a la red durante cuatro segundos. Todas las simulaciones se han realizado en un tiempo de ejecución (*Run time*) de 20s. Para solucionar los problemas de convergencia durante la simulación, es decir, que el incremento temporal caiga por debajo de  $200 \times 10^{-15}$  que es el mínimo valor permitido, se ha forzado al sistema a utilizar incrementos temporales mayores, fijándole el tamaño de paso máximo en 1s.

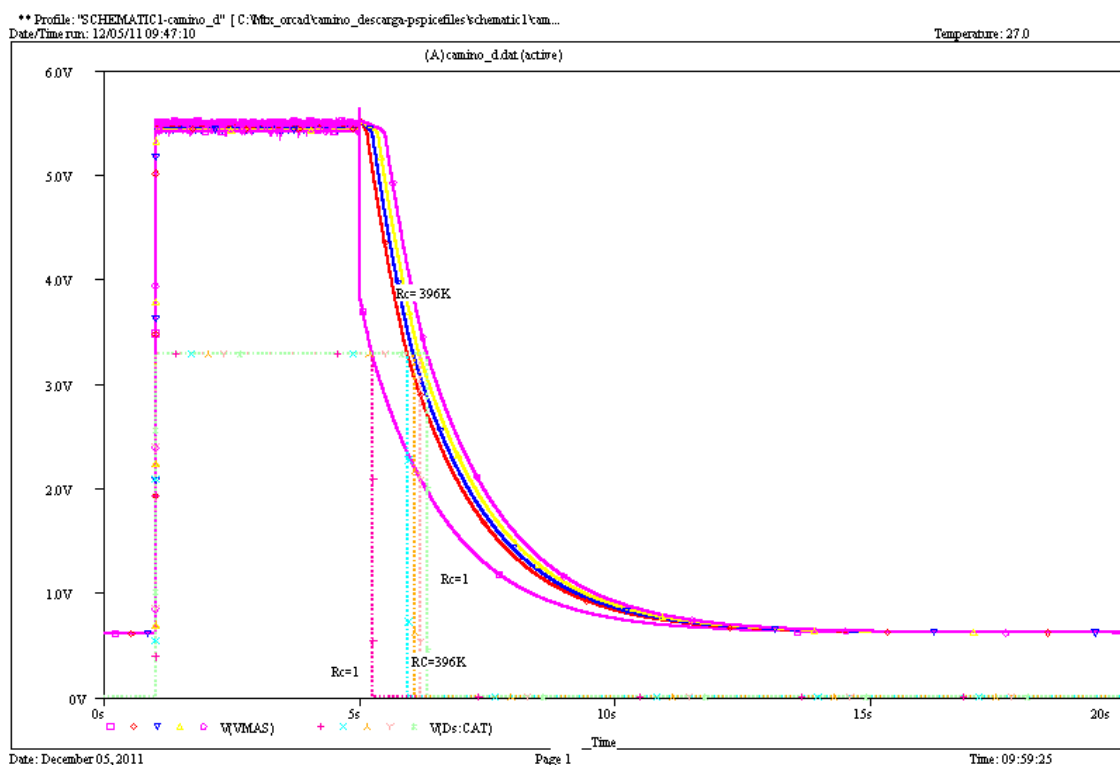


**Ilustración 25: Esquema eléctrico del sensor detector de red**

A la hora de seleccionar el valor del condensador se debe tener en cuenta que el objetivo no es necesariamente obtener una señal perfectamente continua, sino reducir el valor de rizado hasta obtener valores admisibles para el amplificador operacional. Conseguir una señal perfectamente continua exigiría valores de condensadores muy altos del orden de milifaradios, y eso aumentaría considerablemente el consumo, el tamaño y el precio del dispositivo.

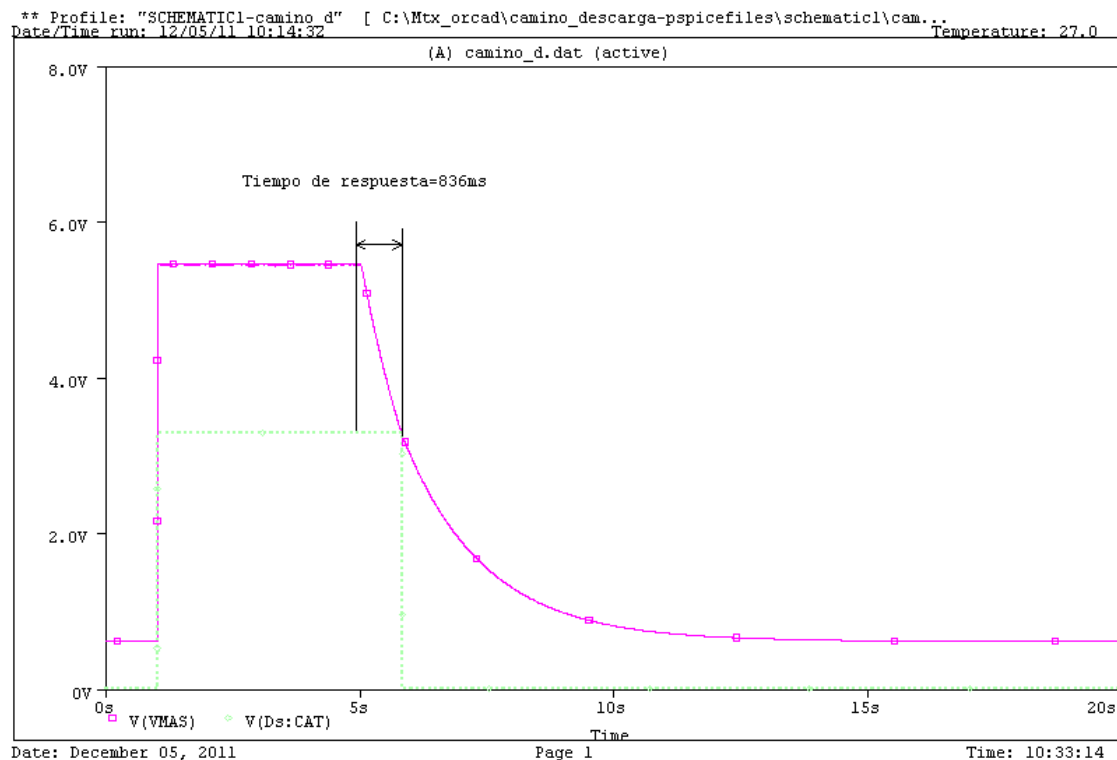
El condensador necesita un camino de descarga de no ser así, una vez realizada la detección de red, el circuito permanecería cargado durante más de 20s [82]. Para agilizar este tiempo a parte de intentar seleccionar un condensador más pequeño se debe incorporar un camino de descarga. Con este objetivo se añade una resistencia  $R_c$  en serie con el condensador y un zener ( $D_m$ ) que trabajará en inversa. Esta es una forma económica de tener una señal apta para el amplificador operacional. Conforme a lo expuesto, el valor del condensador se seleccionó mediante reiteradas simulaciones obteniendo un valor de condensador de 200nF. Un valor de condensador mayor supone disminuir la tensión de rizado pero también aumentar notablemente el tiempo de respuesta de la salida  $V_{outsensored}$ . Entendiendo por tiempo de respuesta el tiempo que transcurre desde que la señal de red desaparece hasta que el sensor de red indica que ha desaparecido. Calculando el valor del condensador como si se tratase de obtener una fuente regulada [81] se obtendría un valor de condensador de 3.2mF con el consecuente aumento del tiempo de respuesta y de consumo adicional, innecesario.

Para calcular el valor adecuado de la resistencia  $R_c$  se hicieron múltiples simulaciones similares a la que aparece en la Ilustración 26, donde las líneas continuas representan los valores de la tensión  $V_{mas}$  para diferentes valores de  $R_c$  entre 1 y 396KΩ y las líneas discontinuas representan el tiempo de respuesta de la salida  $V_{outsensored}$ . De esta gráfica se deduce que ha medida que disminuye el valor de  $R_c$ , el tiempo de respuesta disminuye pero el valor de la tensión de rizado aumenta.

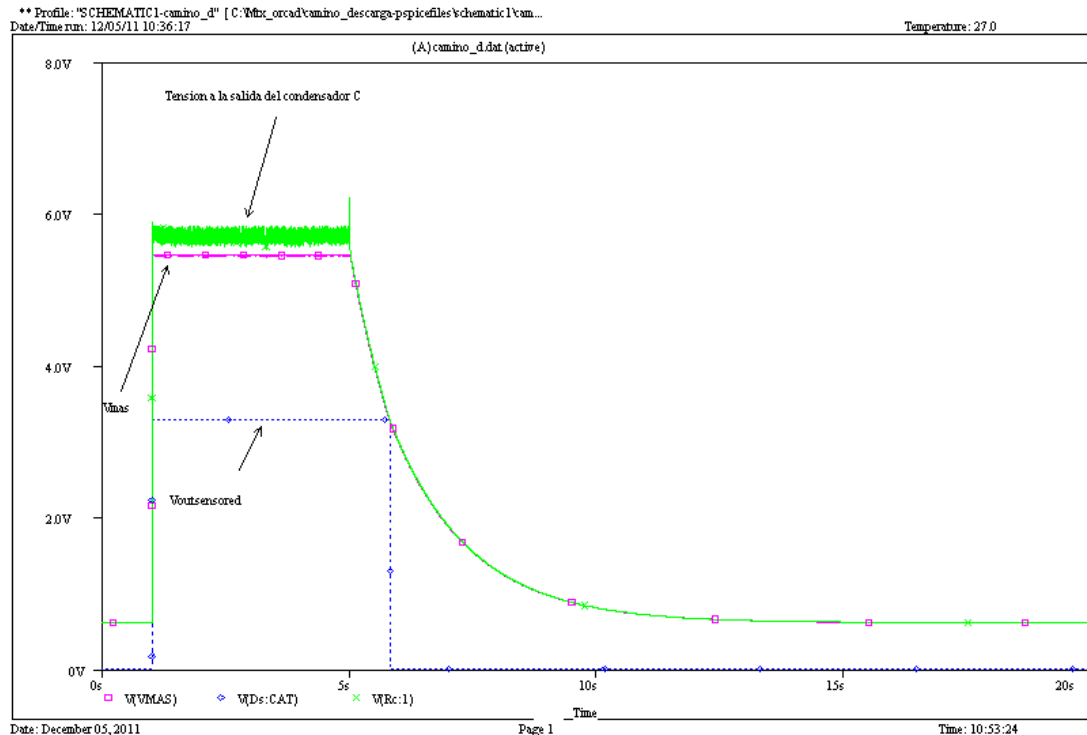


**Ilustración 26: Tiempo de respuesta en función del valor de  $R_c$**

En la Ilustración 27 se muestra como, con un valor de  $R_c$  de  $47K\Omega$ , el tiempo de respuesta corresponde a 836ms y la tensión de rizado se reduce aceptablemente (ver Ilustración 28).

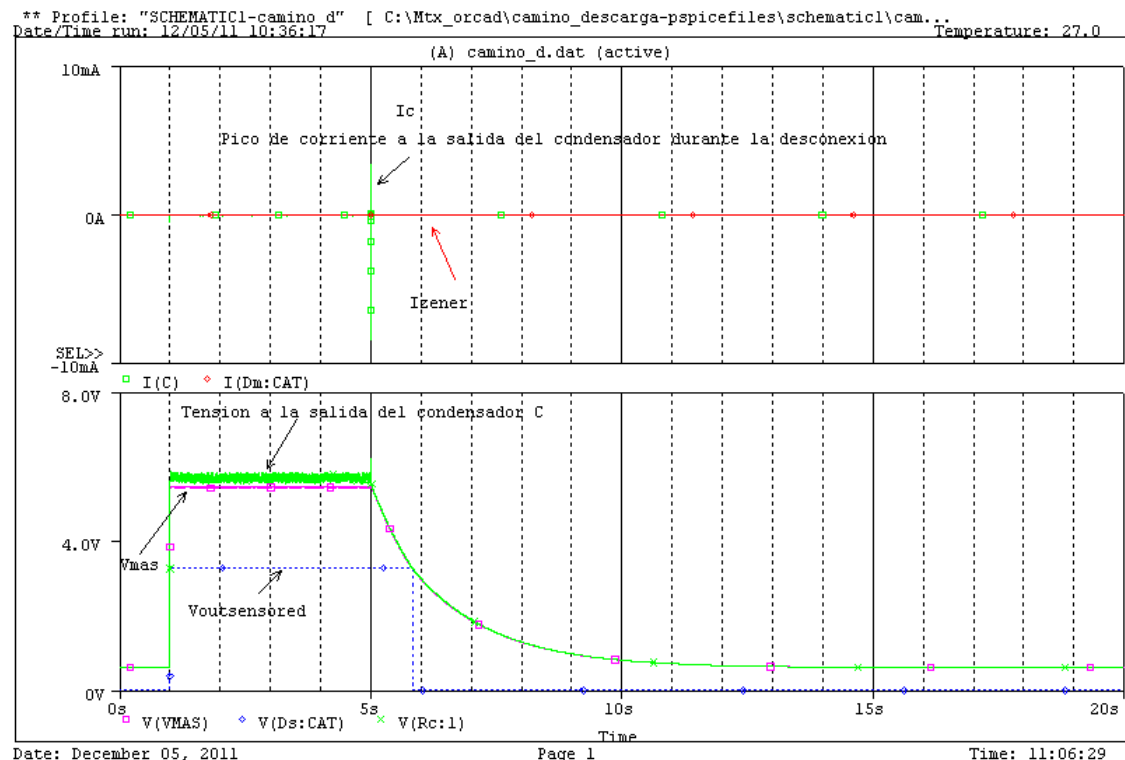


**Ilustración 27: Tiempo de respuesta para  $R_c=47K\Omega$**



**Ilustración 28: Tensión de rizado a la salida del condensador  $R_c=47K\Omega$**

Ante la desconexión no solo aparece un pico de tensión en el condensador absorbido por el zener (ver Ilustración 27) sino que aparece un pico de corriente que también será absorbido por el zener al polarizarse ( Ilustración 29). Con una resistencia de menor valor aumentaría los picos de corriente.



**Ilustración 29: Picos de corriente en la desconexión**

Al incluir a la salida del puente de diodos un condensador de 200nF se obtiene una tensión de salida superior a 5V. El modelo de zener seleccionado corresponde al BZX85C5V6, cuya corriente en el punto de operación,  $I_{zt}$ , es de 45mA y una corriente inversa máxima de 162mA, lo que garantiza que absorba el pico de corriente que aparece ante la desconexión de la red.

## 5.9 BLOQUE DE COMPARACIÓN

La comparación se realiza mediante un amplificador operacional en configuración de comparador no inversor. Si la tensión proporcionada por  $V_{mas}$  es mayor que la tensión de referencia, la salida que proporciona el amplificador operacional será el valor de la alimentación positiva del amplificador. Si por el contrario la tensión proporcionada por  $V_{mas}$  es menor que la tensión de referencia del amplificador la salida será del valor de la alimentación negativa del amplificador. En la Ilustración 30 se puede ver el resumen del funcionamiento del comparador no inversor.



Ilustración 30: Comparador no inversor

Siendo:

- $V_{mas}$  la tensión de entrada acondicionada.
- $V_{ref}$  la tensión de referencia.
- $+V_{cc}$  y  $-V_{cc}$  las tensiones de alimentación del amplificador operacional.

### 5.9.1 Selección del amplificador operacional

Conforme a las restricciones antes detalladas se ha seleccionado el amplificador operacional LM324, la razón principal es su amplio rango de entrada de alimentación (entre 3 y 32V) esto puede permitir reutilizar el sensor en otras aplicaciones futuras. Puesto que el operacional estará configurado como comparador, permitirá un rango variado de salidas (entre 0 y  $V_{cc}-1.5V$ ). También es importante destacar que presenta un consumo de corriente de polarización de entrada muy bajo de unos 45nA compensado con la temperatura. Además de un consumo de corriente muy bajo, de unos 700μA independientemente de la alimentación. Esto último es muy interesante a la hora de no demandar mucha corriente a la batería del vehículo.

En este caso y puesto que la alimentación que proporciona la batería propia del vehículo es doce voltios se alimentará al amplificador entre 0 y 12V.

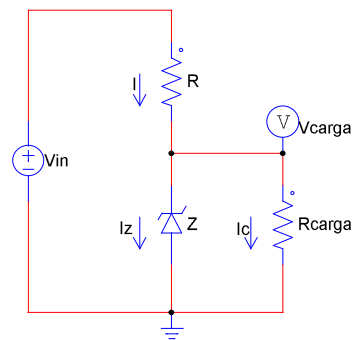
## 5.10 SELECCIÓN DE LA TENSIÓN DE REFERENCIA

La tensión de referencia se fija mediante un divisor de tensión. La alimentación del divisor de tensión será 12V en continua que es la tensión que nos proporciona la batería del vehículo.

Teniendo en cuenta que generalmente la caída de tensión máxima admisible es el 3% de la tensión nominal en instalaciones domésticas y el 5% en las instalaciones industriales. Se ha diseñado el sistema para el peor caso posible. Así una caída del 3 % sobre una alimentación de red de 230V representa una caída de 6,9V es decir una tensión de 223,1V y para 220 una caída de 11V es decir 209V.

Variando la tensión de red ( $V_{red}$ ) entre 230V y 209V se obtiene por simulación en OrCAD los valores de tensión  $V_{mas}$  entre 5.3V y 5.45V para un valor de resistencia  $R_c$  de 200k $\Omega$  respectivamente. La tensión de referencia deberá ser un valor inferior a 5.3V. Se selecciona el diodo zener BZX85C4V3, cuya tensión zener es de 4.3V.

El valor de la resistencia de polarización se determina conforme al circuito de la Ilustración 31, de acuerdo con este circuito se obtiene la ecuación de la resistencia de polarización (9), donde sustituyendo la corriente de polarización en función de la corriente en el zener y de la corriente en la carga (ver ecuación (10)) se obtienen las ecuaciones (11) y (12) para los valores de resistencia de polarización máximos y mínimos respectivamente.



**Ilustración 31: Esquema eléctrico para la polarización de un zener**

$$V_{in} = I \cdot R + V_z \Rightarrow R = \frac{V_{in} - V_z}{I} \quad (9)$$

$$I = I_z + I_c \quad (10)$$

$$R_{max} = \frac{V_{in_{max}} - V_z}{I_{c_{min}} + I_{z_{min}}}$$

**( 11) Valor máximo de la resistencia de polarización de un zener**

$$R_{min} = \frac{V_{in_{min}} - V_z}{I_{c_{max}} + I_{z_{max}}}$$

**( 12) Valor mínimo de la resistencia de polarización de un zener**

Donde:

- $R_{min}$  es el valor mínimo de la resistencia limitadora del zener.
- $R_{max}$  es el valor máximo de la resistencia limitadora del zener.
- $V_{in_{max}}$  es el valor máximo de la tensión de entrada, en este caso de la batería del vehículo.
- $V_{in_{min}}$  es valor mínimo de la tensión de entrada, de la batería del vehículo.
- $V_z$  es la tensión del zener.
- $I_{z_{max}}$  es la máxima intensidad que soporta el zener.
- $I_{z_{min}}$  es la mínima intensidad que soporta el zener.
- $I_{c_{max}}$  es la máxima intensidad que consume la carga.
- $I_{c_{min}}$  es la mínima intensidad que consume la carga.



La carga en el circuito es la entrada al amplificador operacional LM324  $V_{refUA1}$  con una impedancia de entrada muy elevada que permite valores de corriente de carga comprendidos entre 0 y 500nA (valores obtenidos de la hoja de características del LM324 [73])

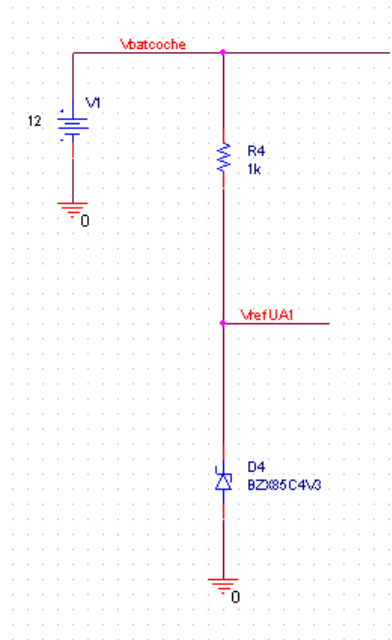
La batería propia del vehículo proporciona una tensión de 12V, así, el valor  $V_{in_{max}}$  y  $V_{in_{min}}$  estará comprendido entre los valores extremos 13 y 11 voltios respectivamente. El valor de la tensión en el diodo zener se ha obtenido en las hojas de características del BZX85C4V3 [72].

Sustituyendo los valores anteriores en las ecuaciones (11) y (12) se obtienen los valores máximos y mínimos de la resistencia de polarización (13) y (14) respectivamente:

$$R_{max} = \frac{V_{batcoche_{max}} - V_z}{I_{refUA1_{min}} + I_{z_{min}}} = \frac{13 - 4.3}{0 + 1m} = 8.7 K\Omega \quad (13)$$

$$R_{min} = \frac{V_{batcoche_{min}} - V_z}{I_{refUA1_{max}} + I_{z_{max}}} = \frac{11 - 4.3}{500n + 50m} = 134\Omega \quad (14)$$

Se selecciona una resistencia de valor comercial de 1k $\Omega$ .



## 5.11 SELECCIÓN DE LA TENSIÓN DE ALIMENTACIÓN DEL OPERACIONAL

La alimentación del amplificador se conecta a la batería de 12V que proporciona el coche. Como ya se ha mencionado se seleccionó el LM324 como amplificador operacional ya que dispone de un amplio rango de tensiones de entrada entre 3 y 32V. El sensor admite una caída de tensión de la tensión de alimentación de más del 66% aunque para el correcto funcionamiento no se recomienda alimentarlo por debajo de 0V.

## 5.12 CIRCUITO ELECTRICO FINAL

De acuerdo con las consideraciones expuestas con anterioridad, el circuito eléctrico quedara definido según la Ilustración 32. Con estos valores la tarjeta TS-7200 interpretará los valores que le proporcione el sensor, de forma que si el sensor detector de red proporciona un uno lógico interpretará que el sistema está conectado a la red y cuando reciba un cero lógico informará al usuario que el sensor no está conectado a la red o bien la batería del coche no está proporcionando tensión.

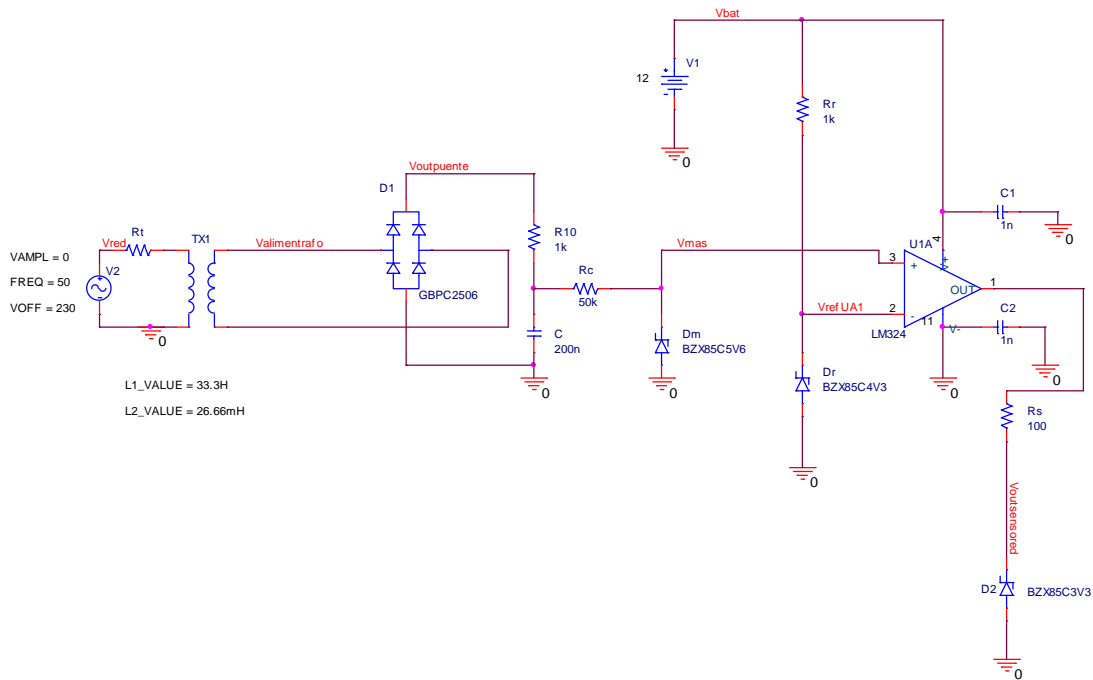


Ilustración 32: Circuito eléctrico completo

## 5.13 ACONDICIONAMIENTO A LA TARJETA TS-7200

Las entradas DIO de la tarjeta TS-7200 llevan incorporadas resistencias *pull up* que mantienen la tensión en 3.3V cuando la entrada no recibe ninguna tensión y en concreto la entrada digital DIO\_0 en la que se conectará la salida del sensor tiene incorporada una resistencia de 4.7K $\Omega$ .

El sensor de red se comporta como un interruptor de manera que para conectar adecuadamente el circuito se incorpora una resistencia a tierra (resistencia *pull down*) en paralelo con la salida del circuito (ver Ilustración 33) que evita que la entrada a DIO\_0 pueda quedar al aire cuando el sensor proporciona un cero lógico. De no incorporar una conexión a tierra cuando el sensor proporcionara un cero lógico, la lectura de la tarjeta TS-7200 sería de un uno lógico. El valor de esta resistencia debe hacer que el valor de la entrada DIO\_0 sea inferior a 1,3V para que la entrada sea interpretada como un cero, se ha seleccionado una resistencia de valor comercial de 100 $\Omega$ .

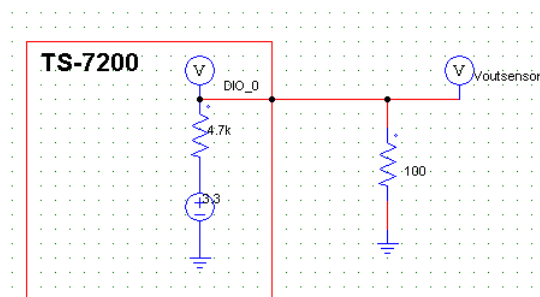


Ilustración 33: Acondicionamiento de la entrada de la tarjeta TS-7200

## 6 SENSOR DE CORRIENTE

### 6.1 OBJETIVO

Existe la necesidad de medir la corriente que se obtiene de la batería y proporcionar esta información a la placa TS-7200.

### 6.2 RESTRICCIONES

La tarjeta TS-7200 dispone de un convertidor A/D de 12 bits con un rango de tensiones de entrada entre 0 y 3.3V.

### 6.3 MEDIDA DE LA CORRIENTE

Para medir la corriente que se obtiene de una batería se puede colocar una pequeña resistencia de carga en serie con las cargas. En ella se puede medir la tensión y acondicionar su valor. Esta forma de medir es sencilla aunque poco práctica ya que trabajar con corrientes tan elevadas puede ser peligroso. Realizar esta medida de forma segura es fundamental, por ello, se ha recurrido a un transductor de corriente.

### 6.4 TRANSDUCTORES DE CORRIENTE BASADOS EN EL EFECTO HALL

Un transductor de corriente es un elemento que transforma la corriente proporcionada por la batería en una tensión proporcional a la corriente de entrada.

Tanto los transductores en bucle abierto como en bucle cerrado se basan en el efecto Hall. De acuerdo con el efecto Hall, cuando por una lámina por la que circula corriente  $I$ , atraviesa un campo magnético  $B$  perpendicular al movimiento de las cargas, produce una diferencia de potencial  $V_H$  en la lámina perpendicular al movimiento de las cargas y al campo magnético. A esta diferencia de potencial  $V_H$  se la denomina *Voltaje Hall* ( $V_H$ ) y su valor viene determinado por la ecuación (15):

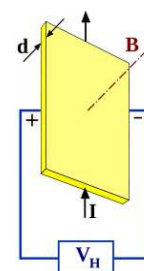


Ilustración 34:  
Efecto Hall

$$V_H = \left( \frac{K}{d} \right) \cdot I_c \cdot B \quad (15)$$

Donde K es la constante de Hall del material usado, d el espesor de la lámina, B el campo magnético que la atraviesa e  $I_c$  es la corriente que circula por la lámina.

En un transductor en bucle abierto basado en el efecto Hall, la corriente en el primario ( $I_p$ ) al atravesar el campo magnético generado por una bobina, produce un incremento de la tensión de Hall. Por otro lado  $I_c$  está alimentado por una fuente de corriente constante (ver Ilustración 35).

En la región lineal del ciclo de histéresis el campo magnético B, es proporcional a la corriente en el primario según la ecuación (16) [46].

$$B_{gap\_aire} = cte(a) \cdot I_p \quad (16)$$

Excepto  $I_c$  todos los términos de la ecuación (17) son constantes, y se pueden agrupar en una constante (b) (ver ecuación (18)).

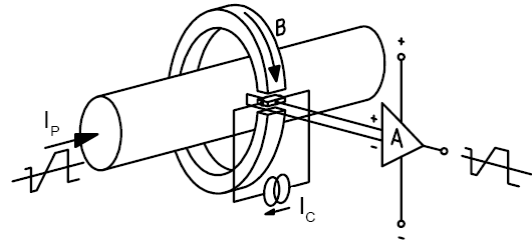
$$V_H = \left( \frac{K}{d} \right) \cdot cte(a) \cdot I_c \quad (17) \quad \Rightarrow \quad V_H = cte(b) \cdot I_c \quad (18)$$

La medida del Voltaje Hall se amplifica antes de proporcionársela al usuario en forma de tensión. Los transductores en bucle abierto basados en el efecto Hall miden tanto corrientes continuas como corrientes alterna.

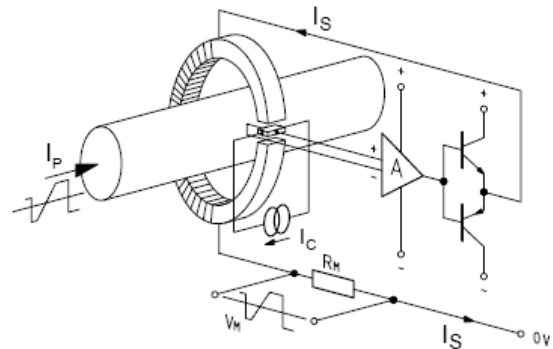
Un transductor en bucle cerrado, basado en el efecto Hall (ver Ilustración 36), incluye un circuito integrado de compensación que mejora el rendimiento del transductor. El funcionamiento es básicamente el mismo que en bucle abierto. La corriente en el primario ( $I_p$ ) fluye a través del conductor creando un flujo magnético primario ( $B_p$ ). El bucle cerrado se realimenta con la corriente secundaria  $I_s$  que crea una inducción secundaria  $B_s$  para compensar la inducción en el primario ( $B_p$ ) creada por la corriente  $I_p$ . Lógicamente,  $I_s$  es mucho menor que  $I_p$ .

El transductor en bucle cerrado mide al igual que el de bucle abierto tanto en corriente alterna como en corriente continua. La corriente en el secundario es proporcional en todo momento a la corriente en el primario cumpliéndose en todo la relación de transformación dada por la ecuación(19):

$$N_p \times I_p = I_s \times N_s \quad (19)$$



**Ilustración 35: Transductor en bucle abierto**



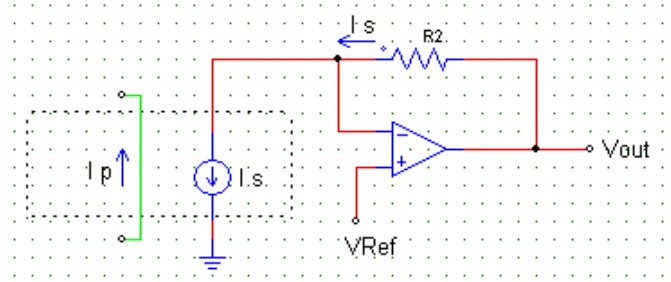
**Ilustración 36: Transductor en bucle cerrado**

Donde  $N_s$  y  $N_p$  corresponden al número de espiras del secundario y del primario respectivamente.  $N_p$  es igual a 1, puesto que el conductor es el que genera el campo magnético, lo que equivale a una única espira [46]. Sustituyendo el valor de  $N_p$  en la ecuación (19) se obtiene el valor de la corriente en el secundario, ecuación (20):

$$I_s = \frac{I_p}{N_s} \quad (20)$$

El transductor contiene un amplificador con realimentación negativa de manera que se puede decir que  $V_- \cong V_+ = V_{ref}$ , siendo  $V_{ref}$ , la tensión de referencia. La corriente  $I_s$ , vendrá entonces, definida por la ecuación (21):

$$I_s = \frac{V_{out} - V_{ref}}{R_{IM}} \quad (21)$$



**Ilustración 37: Esquema básico de un transductor LTS**

Despejando  $V_{out}$  en la ecuación (21) y sustituyendo el valor de  $I_s$  por el de la ecuación (20), da como resultado la ecuación (22), donde se puede observar que existe una relación lineal entre la tensión de salida  $V_{out}$  y la corriente en el primario  $I_p$ .

$$V_{out} = R_{IM} \cdot I_s + V_{ref} \Rightarrow V_{out} = R_{IM} \cdot \frac{I_p}{N_s} + V_{ref} \quad (22)$$

La curva  $V_{out}$  se define por los parámetros  $a$  y  $b$  tal y de acuerdo con las ecuaciones (23), (24) y (25). Donde  $a$  y  $b$  son constantes que proporciona el fabricante. El valor de  $I_p$  estará dentro del rango ( $I_p$ ,  $-I_p$ ) en función del sentido en el que circule la corriente en el primario.

$$V_{out} = a + bI_p \quad (23) \quad \left\{ \begin{array}{l} a = V_{ref} \quad (24) \\ b = \frac{R_{IM}}{N_s} \quad (25) \end{array} \right.$$

## 6.5 SELECCIÓN DEL TRANSDUCTOR DE CORRIENTE

En la actualidad existen en el mercado circuitos integrados que realizan esta función. La serie LTS de la marca LEM proporciona transductores de corrientes en una gama amplia de corrientes nominales. El transductor de corriente continua, utilizado, está basado en el efecto Hall en bucle cerrado. Permitiendo la medida directa de la corriente mediante un aislamiento galvánico entre el primario y secundario. Esto garantiza que exista una distancia física entre el circuito conectado a la batería y el circuito de medida. Lo que garantiza una mayor seguridad de los usuarios.

El transductor se deberá conectar a la batería en serie con las cargas, lo que representa una de las principales ventajas de este transductor al no introducir un voltaje en serie adicional al circuito que está midiendo.

## 6.6 TRANSDUCTOR SELECCIONADO

### 6.6.1 Características que determinan el transductor seleccionado

#### 6.6.1.1 Principio de funcionamiento:

En la Ilustración 38 aparece el esquema de un LTS 15-NP. Este transductor presenta dos terminales de entrada que deberán conectarse a la batería y en serie con las cargas. Se alimenta a 5 V y su precisión es de  $\pm 0,2\%$  del valor real de la corriente.

El transductor LTS-15-NP convierte la corriente eléctrica a medir, en una señal de tensión correspondiente a la recta (26) donde  $I_{PN}$  es la corriente nominal en el primario. De manera, que a una corriente de valor nulo le corresponde un nivel de tensión de 2,5V como se puede observar en la gráfica de la Ilustración 39. Esta gráfica, proporcionada por el fabricante, representa la función de transferencia del transductor, es decir, la relación entre la entrada y la salida del transductor.

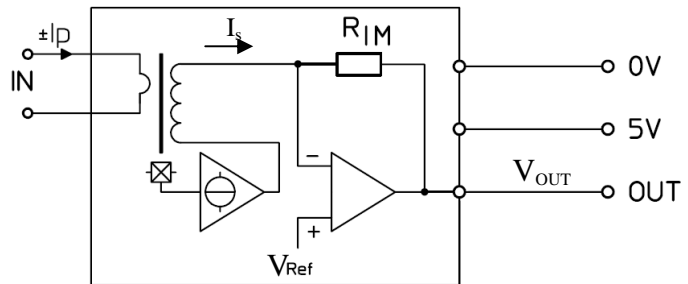


Ilustración 38 Esquema del transductor de corriente

$$2,5 \pm (0,625 \cdot \frac{I_P}{I_{PN}}) \quad (26)$$

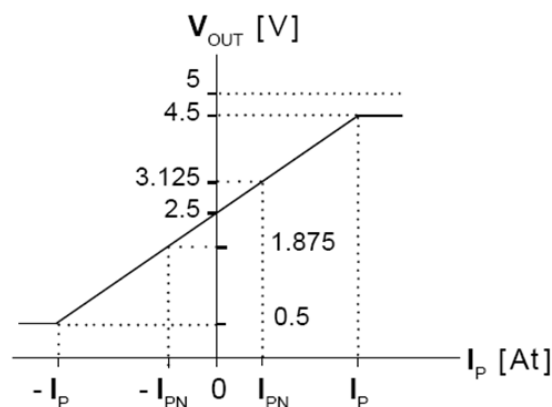


Ilustración 39: Relación entre la entrada y la salida del transductor de corriente

El fabricante seleccionado dispone de una amplia gama de transductores. Para seleccionar uno de ellos, en primer lugar será necesario conocer la máxima corriente demandada por las cargas ya que la selección del transductor se hará conforme a las corrientes con las que va a trabajar.

#### 6.6.1.2 Corriente demandada por las cargas

Los equipos críticos están conectados a la SAI (ver página 61). La batería adicional alimenta un máximo de cuatro ordenadores, cada ordenador consume como máximo 110W

(dato proporcionado por el fabricante) y está alimentado a 230V. Así la máxima corriente demanda por cada ordenador será de 0,5A según la ecuación (27), por tanto cuatro ordenadores demandarán una corriente eficaz de 2A (28).

$$I_{ef} = \frac{P}{V_{ef}} = \frac{110W}{230V} = 0,5A \quad (27)$$

$$I_{ef\_pc} = 0,5 \cdot 4 = 2A \quad (28)$$

Además podrán estar conectados a la batería el Router, y los ventiladores:

Router	$I_{ef} = \frac{P}{V_{ef}} = \frac{20}{230} = 0,086A$
Ventiladores	$I_{ef} = \frac{P}{V_{ef}} = \frac{30}{230} = 0,130A$

La corriente eficaz total máxima que demandarán las cargas vendrá dada por la ecuación (29). A la hora de diseñar se ha tenido en cuenta que no siempre el uso final se corresponde con el que se pensó inicialmente, además del hecho de que probablemente terminen por incorporarse nuevas aplicaciones, de ahí que se sobredimensiona el posible consumo. Seleccionamos finalmente el LTS-15, que nos permite un amplio rango de medida entre 0 y 45 amperios, ajustable a una corriente nominal de trabajo de entre  $\pm 5$  y  $\pm 15A$ . Y cuya tensión de salida se define mediante la ecuación (30).

$$I_{ef\_TOT} = 2 + 0,086 + 0,130 = 2,216A \quad (29)$$

$$2,5 \pm \left(0,625 \cdot \frac{I_P}{I_{PN}}\right) \quad (30)$$

$I_{PN}$	Corriente nominal en el primario	15	At
$I_P$	Gama de medición de la corriente principal	$0.. \pm 45$	At
$V_{OUT}$	Tensión de salida analógica	$2.5 \pm \left(0.625 \cdot \frac{I_P}{I_{PN}}\right)$	V
$R_L$	Resistencia de carga	$\geq 2k$	$\Omega$
$V_C$	Tensión de alimentación	5	V
f	Frecuencia del ancho de banda	DC..100	KHz

**Tabla 8: Características del LTS-15NP**

En la Tabla 8 aparecen detalla las principales características del transductor LTS-15NP. Las razones por las que se selecciona este transductor son:

- El límite de intensidad nominal de funcionamiento del transductor es configurable, existiendo tres posibilidades:  $\pm 15A$ ,  $\pm 7.5A$  y  $\pm 5A$ .
- El transductor puede medir la corriente en ambos sentidos.

Number of primary turns	Primary nominal current rms $I_{PN}$ [A]	Nominal output voltage $V_{OUT}$ [V]	Primary resistance $R_P$ [mΩ]	Primary insertion inductance $L_P$ [μH]	Recommended connections
1	$\pm 15$	$2.5 \pm 0.625$	0.18	0.013	
2	$\pm 7.5$	$2.5 \pm 0.625$	0.81	0.05	
3	$\pm 5$	$2.5 \pm 0.625$	1.62	0.12	

Tabla 9: Tabla proporcionada por el fabricante del LTS 15NP

De acuerdo con la tabla proporcionada por el fabricante (ver Tabla 9) se seleccionó la configuración correspondiente a una corriente nominal de  $\pm 5A$ . El conversor analógico digital de la tarjeta TS-7200 funciona en un rango de entrada de entre 0 y 3.3V. Por tanto será necesario realizar un acondicionamiento de la salida que modifique el rango del transductor de entre 0 y 5V a un rango que admita la tarjeta controladora, es decir, entre 0 a 3.3V.

## 6.7 BLOQUE ACONDICIONADOR

El bloque acondicionador consiste principalmente en la construcción de un amplificador de instrumentación a partir de un amplificador operacional. Los amplificadores de instrumentación no son más que dispositivos concebidos para la amplificación de señales de tensión diferencial

Para facilitar una primera aproximación la salida del sensor de corriente se simuló como una fuente de tensión continua con un rango de valores idénticos al que proporciona el transductor. La conversión de corriente a tensión se realiza mediante el transductor de corriente, como ya se ha explicado en el apartado 6.4, que se sitúa en serie con la batería y las cargas.

### 6.7.1 Ganancia del amplificador de instrumentación

Aplicando el *Teorema de superposición* (ecuación 31) al esquema eléctrico de la Ilustración 40 se obtiene el valor de la tensión en el sensor de corriente,  $V_{sensocorriente}$ , en función de  $V_{transductor}^-$  y  $V_{UA1}^+$  (32) en la que se sustituye el valor de la tensión en el terminal positivo (33) del amplificador operacional,  $V_{UA1}^+$  en función de  $V_{transductor}^+$  (ecuaciones (34) y (35)).

$$V_{sensocorriente} = V_{sensocorriente} \Big|_{V_{UA1}^+ = 0} + V_{sensocorriente} \Big|_{V_{transductor}^- = 0} \quad (31)$$

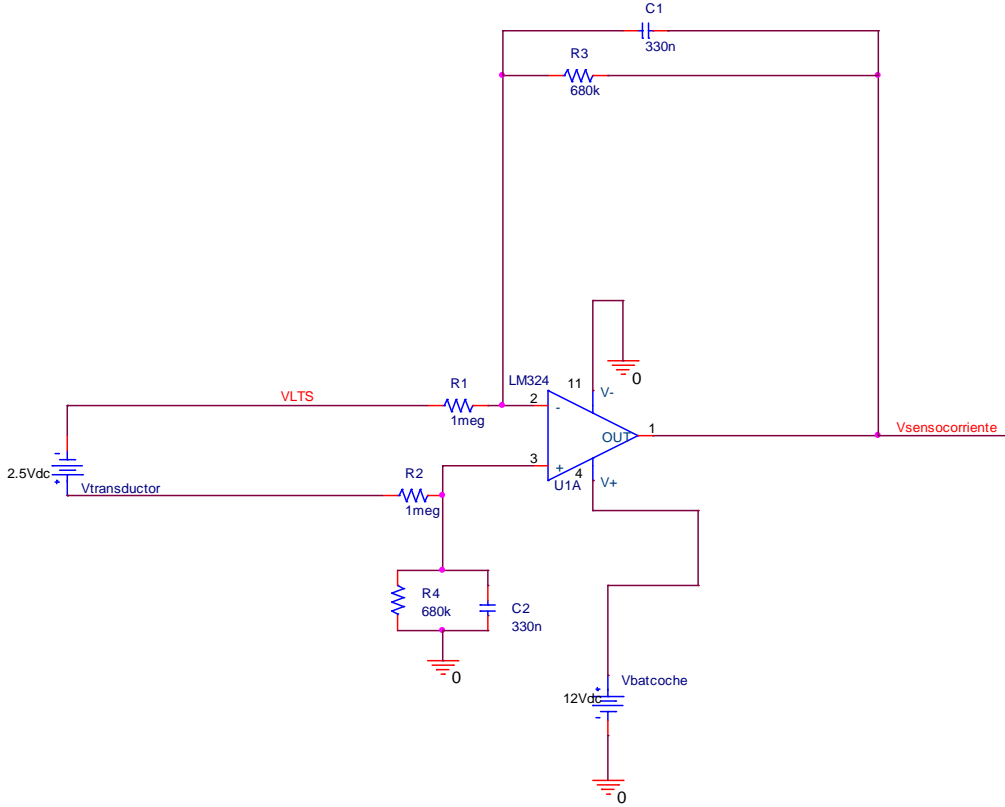
$$V_{sensocorriente} = -\frac{R_3}{R_1} V_{transductor}^- + \left(\frac{R_3}{R_1} + 1\right) \cdot V_{UA1}^+ \quad (32)$$

$$V_{UA1}^+ = \frac{R_4}{R_2 + R_4} \cdot V_{transductor}^+ \quad (33)$$



$$V_{sensocorriente} = -\frac{R_3}{R_1} V_{transductor}^- + \left( \left( \frac{R_3}{R_1} + 1 \right) \cdot \frac{R_4}{R_2 + R_4} \cdot V_{transductor}^+ \right) = \quad (34)$$

$$= -\frac{R_3}{R_1} V_{transductor}^- + \left( \left( \frac{R_3 + R_1}{R_1} \cdot \frac{R_4}{R_2 + R_4} \cdot V_{transductor}^+ \right) \right) \quad (35)$$



**Ilustración 40: Esquema eléctrico sensor de corriente**

Si se seleccionan los valores de las resistencias tal que  $R_2$  sea igual a  $R_1$  y  $R_4$  igual a  $R_3$  se puede desarrollar la ecuación (36) conforme a las ecuaciones (37).

$$\left. \begin{array}{l} R_2 = R_1 \\ R_4 = R_3 \end{array} \right\} \Rightarrow V_{sensocorriente} = -\frac{R_3}{R_1} V_{transductor}^- + \left( \left( \frac{R_3 + R_1}{R_1} \cdot V_{UA1}^+ \cdot \frac{R_3}{R_1 + R_3} \cdot V_{transductor}^+ \right) \right) \Rightarrow \quad (36)$$

$$\Rightarrow V_{sensocorriente} = -\frac{R_3}{R_1} V_{transductor}^- + \frac{R_3}{R_1} \cdot V_{transductor}^+ \quad (37)$$

Si consideramos que la ganancia del sistema es la tensión a la salida entre la tensión de entrada, la ganancia se puede expresar en función de las resistencias, conforme a la ecuación (38).

$$G = \frac{V_{sensocorriente}}{V_{transductor}^+ + V_{transductor}^-} = \frac{R_3}{R_1} = \frac{R_4}{R_2} \quad (38)$$

$$G = \frac{R_3}{R_1} = \frac{R_4}{R_2} \quad (39)$$

Sustituyendo en la ecuación (39) el valor máximo de la tensión que proporciona el transductor son 4.5V y el valor máximo que admite el conversor analógico-digital de la tarjeta TS-7200 son 3.3V se obtiene una ganancia máxima de 0.73V (41). Por tanto, el valor de la ganancia del acondicionamiento deberá ser inferior a 0.73V.

$$G_{\max} = \frac{V_{\text{sensocorriente}}_{\max}}{V_{\text{transductor}}_{\max}} = \frac{3,3V}{4,5V} = 0,73 \frac{V}{V} \quad (40)$$

Como se ha demostrado los pares de resistencias R2, R4 y R3, R1, fijan la ganancia. Seleccionando las resistencias con valores comerciales de forma que la ganancia sea inferior a los 0.73V calculados, se obtiene una ganancia de 0.68V/V en la ecuación (42).

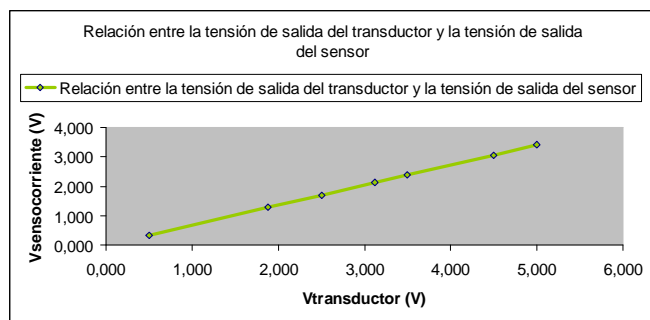
$$G = \frac{R_3}{R_1} = \frac{R_4}{R_2} = \frac{680k\Omega}{1M\Omega} = 0.68 \frac{V}{V} \quad (41)$$

Las resistencias R1 y R11 limitan la corriente de entrada, de ahí que se hayan elegido unos valores de resistencia elevados. C1 y C2 limitan el ancho de banda. El amplificador seleccionado es de nuevo el LM324, que como ya se ha mencionado en este documento entre sus principales características se encuentran la de alimentación variable.

La relación entre la entrada y la salida del sensor obtenida mediante simulación en OrCAD del esquema eléctrico, da como resultado la Tabla 10, que representada gráficamente la Ilustración 41. El acondicionamiento de la salida del transductor es lineal a la tensión de salida del transductor como se aprecia en la ilustración.

V <sub>transductor</sub> (V)	V <sub>sensocorriente</sub> (V)
5	3.403
4.5	3.063
3.5	2.383
3.125	2.128
2.5	1.704
1.875	1.279
0.5	0.344

**Tabla 10:** Relación entre la tensión de salida del transductor y la tensión de salida del sensor después del acondicionamiento

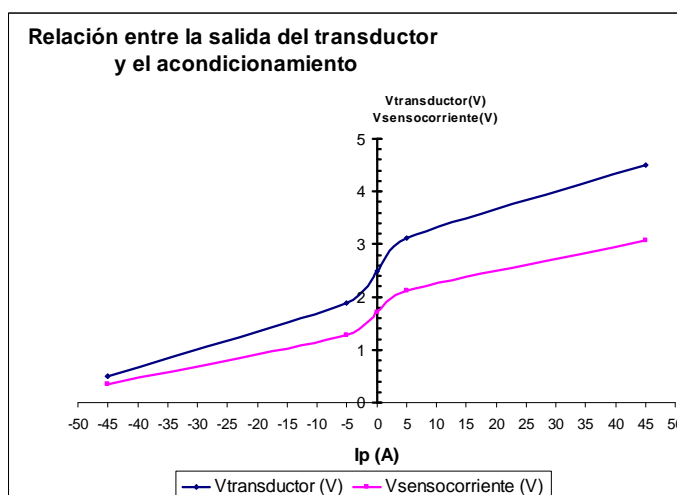


**Ilustración 41:** Relación entre la tensión de salida del transductor y la tensión de salida del sensor

Una vez conocida la relación del transductor entre la tensión del sensor de corriente se puede calcular la relación entre la Ip (corriente demanda a la batería) y la tensión en el sensor de corriente. En la Tabla 11, la primera columna, V<sub>transductor</sub>(V) y la columna de la corriente demanda por la batería I(A) son datos proporcionados por el fabricante [47], la columna V<sub>sensocorriente</sub>(V) son los datos que se habían calculado mediante la simulación en OrCAD. Por último la columna ΔV es la diferencia entre V<sub>transductor</sub> y V<sub>sensocorriente</sub> que representa la desviación que se produce en el acondicionamiento de la señal.

$V_{\text{transductor}}$ (V)	$V_{\text{sensocorriente}}$ (V)	$\Delta V$ (V)	$I_p$ (A)
4.5	3.063	1,437	45
3.125	2.128	0,997	5
2.5	1.704	0,796	0
1.875	1.279	0,596	-5
0.5	0.344	0,156	-45

**Tabla 11: Relación de unidades entre el transductor de corriente la tensión en el sensor de corriente y la corriente demandada por la batería**

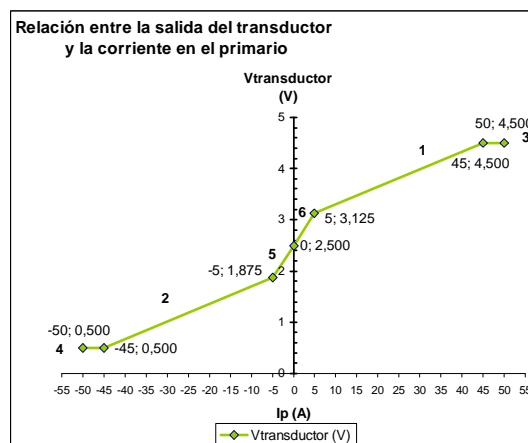


**Ilustración 42: Curvas de funcionamiento del transductor y del sensor de corriente**

### 6.7.2 Relación entre la salida del transductor y el acondicionamiento

Con los resultados de la Tabla 11, se construye la gráfica de la Ilustración 42. Como se puede observar la tensión a la salida del sensor de corriente tiene un comportamiento similar al transductor de corriente pero el rango de tensiones de la salida se ajusta al rango de funcionamiento del conversor analógico-digital (entre 0 y 3.3V) de la tarjeta TS-7200. A la hora de programar es necesario interpretar el dato que recibe el conversor analógico-digital de la tarjeta TS-7200. Si éste recibe 1.407V debe interpretar que el transductor tiene una salida de 2.5V o lo que es lo mismo que la corriente demanda en la batería son 0A.

Con el objetivo de hacer posible la interpretación de los datos, se ha modelado la curva del transductor de corriente mediante una *suavización por splines* [78]. De nuevo mediante la Tabla 11 al realizar la gráfica entre la corriente demanda por la batería  $I_p$  y la tensión en el transductor  $V_{\text{transductor}}$  se obtiene la gráfica que aparece en la Ilustración 43, donde se pueden apreciar seis tramos de pendiente. El objetivo de los cálculos que aparecen a continuación será el de definir la ecuación de la curva para los diferentes tramos de pendiente.



**Ilustración 43: Splines de modelado**

Los splines 1, 2, 5 y 6 se han calculado tomando los puntos de inicio y final de cada tramo. En los splines 3 y 4, la tensión en el transductor es constante e independiente de la corriente demanda por la batería. La curva de funcionamiento conformada por los splines será la que se utilice en la programación.

**Esplín 1:**

$$\begin{aligned} 3.125 &= a + 5 \cdot b \\ 4.5 &= a + 45 \cdot b \end{aligned} \quad \boxed{\begin{aligned} b &= 0.343 \\ a &= 2.953 \end{aligned}} \quad I_p = \frac{V_{\text{transductor}} - a}{b} = \frac{V_{\text{transductor}} - 2.953}{0.343} \quad (42)$$

**Esplín 2:**

$$\begin{aligned} 1.875 &= a + (-5) \cdot b \\ 0.5 &= a + (-45) \cdot b \end{aligned} \quad \boxed{\begin{aligned} b &= 0.343 \\ a &= 2.046 \end{aligned}} \quad I_p = \frac{V_{\text{transductor}} - a}{b} = \frac{V_{\text{transductor}} - 2.046}{0.343} \quad (43)$$

**Esplín 3:**  $V_{\text{transductor}} = 4.5$  (44)

**Esplín 4:**  $V_{\text{transductor}} = 0.5$  (45)

**Esplín 5:**

$$\begin{aligned} 1.875 &= a + (-5) \cdot b \\ 2.5 &= a + (0) \cdot b \end{aligned} \quad \boxed{\begin{aligned} b &= 0.125 \\ a &= 2.5 \end{aligned}} \quad I_p = \frac{V_{\text{transductor}} - a}{b} = \frac{V_{\text{transductor}} - 2.5}{0.125} \quad (46)$$

**Esplín 6:**

$$\begin{aligned} 3.125 &= a + 5 \cdot b \\ 2.5 &= a + (0) \cdot b \end{aligned} \quad \boxed{\begin{aligned} b &= 0.125 \\ a &= 2.5 \end{aligned}} \quad I_p = \frac{V_{\text{transductor}} - a}{b} = \frac{V_{\text{transductor}} - 2.5}{0.125} \quad (47)$$

Después se han utilizado los splines en la programación de la tarjeta TS-7200. La curva modelada se programa mediante *if* anidados donde,  $V_{tc}$  es la salida del transductor de corriente.  $I_p$  es la corriente demanda por las cargas. En la Ilustración 44 aparece un extracto de la programación en C de la curva que describe la Ilustración 43.

```
if (Vtc > 4.5)
    Ip = 45;
else
    if (Vtc > 3.125)
        Ip = (Vtc - 2.953125) / 0.034375;
    else
        if (Vtc > 1.875)
            Ip = (Vtc - 2.5) / 0.125;
        else
            if (Vtc > 0.5)
                Ip = (Vtc - 2.046875) / 0.034375;
            else
                Ip = -45;
```

**Ilustración 44:** Extracto de la programación en C de la curva del transductor

Una vez modelada la relación entre la corriente demanda por las cargas  $I_p$  y la tensión a la salida del transductor  $V_{\text{transductor}}$  faltaría por modelar la relación entre la salida del transductor de corriente y la entrada a la tarjeta TS-7200  $V_{\text{sensocorriente}}$ .

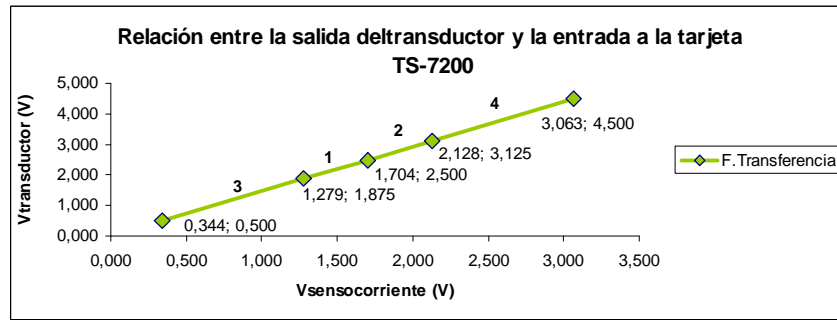


Ilustración 45

Aunque en la Ilustración 45 se aprecia que la relación entre las tensiones  $V_{\text{sensocorriente}}$  y  $V_{\text{transductor}}$  es prácticamente lineal, se ha modelado curva mediante esplines, con los valores se han obtenido de la Tabla 11, para una mayor precisión en el resultado. Desarrollando los esplines análogamente a la conversión anterior se obtienen los cuatro tramos (49), (50), (51), (52). De nuevo la curva modelada se programará mediante *if* anidados (ver Ilustración 46).

Esplín 1:

$$1.875 = a + 1.279 \cdot b$$

$$2.5 = a + 1.704 \cdot b$$

$$b = 1.4705$$

$$a = -0.0058$$

( 48 )

$$V_{\text{transductor}} = a + b \cdot V_{\text{sensocorriente}} = -0.0058 + 1.47V_{\text{sensocorriente}}$$

Esplín 2:

$$3.125 = a + 2.128 \cdot b$$

$$2.5 = a + 1.704 \cdot b$$

$$b = 1.4705$$

$$a = -0.0117$$

( 49 )

$$V_{\text{transductor}} = a + b \cdot V_{\text{sensocorriente}} = -0.0117 + 1.47V_{\text{sensocorriente}}$$

Esplín 3:

$$1.875 = a + 1.279 \cdot b$$

$$0.5 = a + 0.3444 \cdot b$$

$$b = 1.4705$$

$$a = -0.0058$$

( 50 )

$$V_{\text{transductor}} = a + b \cdot V_{\text{sensocorriente}} = -0.0058 + 1.47V_{\text{sensocorriente}}$$

Esplín 4:

$$3.125 = a + 2.128 \cdot b$$

$$4.5 = a + 2.128 \cdot b$$

$$b = 1.4705$$

$$a = -0.0044$$

( 51 )

En la Ilustración 46 puede verse un ejemplo de código necesario a implementar por el maestro cuando desee conocer la corriente demandada a la batería por las cargas a partir de la lectura que realiza el conversor analógico digital de la tarjeta TS-7200. Donde,  $V_{\text{tc}}$  es la salida del transductor de corriente,  $\text{voltADC0}$  es la variable que almacena la salida del transductor de corriente  $V_{\text{sensocorriente}}$ , es decir la entrada de la tarjeta TS-7200,  $V_{\text{tc}}$  es la salida del transductor de corriente e  $I_p$  es la corriente demanda por las cargas.

```

if (voltADC0 >= 2.128)
    Vtc = (-0.004 + 1.47 * voltADC0);
else
    if (voltADC0 >= 1.704)
        Vtc = (-0.011 + 1.47 * voltADC0);

    else
        if (voltADC0 >= 1.279)
            Vtc = (-0.004 + 1.47 * voltADC0);
        else
            Vtc = (-0.005 + 1.47 * voltADC0);

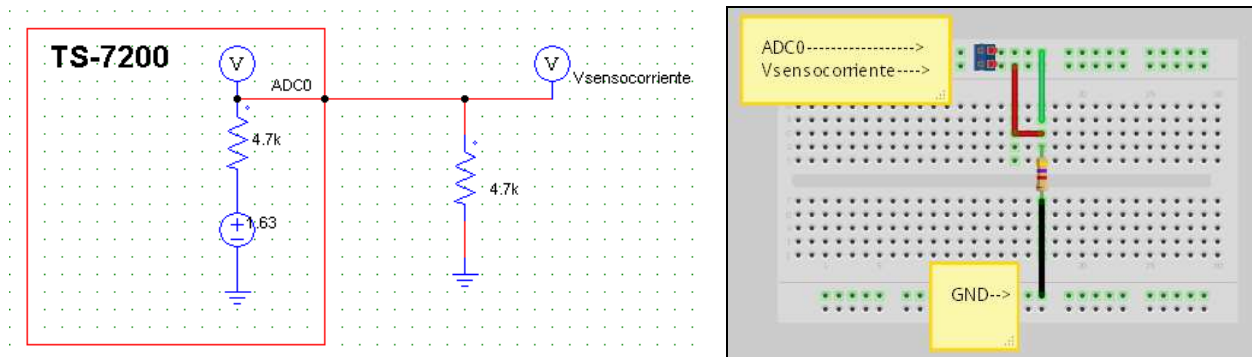
if (Vtc > 4.5)
    Ip = 45;
else
    if (Vtc > 3.125)
        Ip = (Vtc - 2.953125) / 0.034375;
    else
        if (Vtc > 1.875)
            Ip = (Vtc - 2.5) / 0.125;
        else
            if (Vtc > 0.5)
                Ip = (Vtc - 2.046875) / 0.034375;
            else
                Ip = -45;

```

**Ilustración 46:** Extracto de la programación de la relación entre la salida del transductor y la entrada de la tarjeta TS7200

## 6.8 ADAPTACIÓN DEL SENSOR A LA PLACA TS-7200

Al ser un circuito digital se ha utilizado una resistencia a la salida para desacoplarlo, ya que los condensadores introducen una gran carga capacitiva que puede provocar inestabilidad. El valor de esta resistencia se ha seleccionado de manera experimental en 4.7KΩ colocada según la Ilustración 47, ya que su valor puede variar según el conversor analógico digital del que se trate. Permite una correcta lectura del ADC de la tarjeta TS-7200 en todo momento siendo el error que introduce despreciable.



**Ilustración 47:** Resistencia adicional

## 7 ESTIMACIÓN DE LA CAPACIDAD DE UNA BATERÍA

Como ya se había mencionado antes en el apartado 4.4.3 la capacidad de una batería se puede estimar midiendo la tensión entre los bornes de la misma. El circuito de la Ilustración 48 se ha diseñado un circuito basándose en el circuito del apartado 6.7 donde ya se explicó detalladamente.

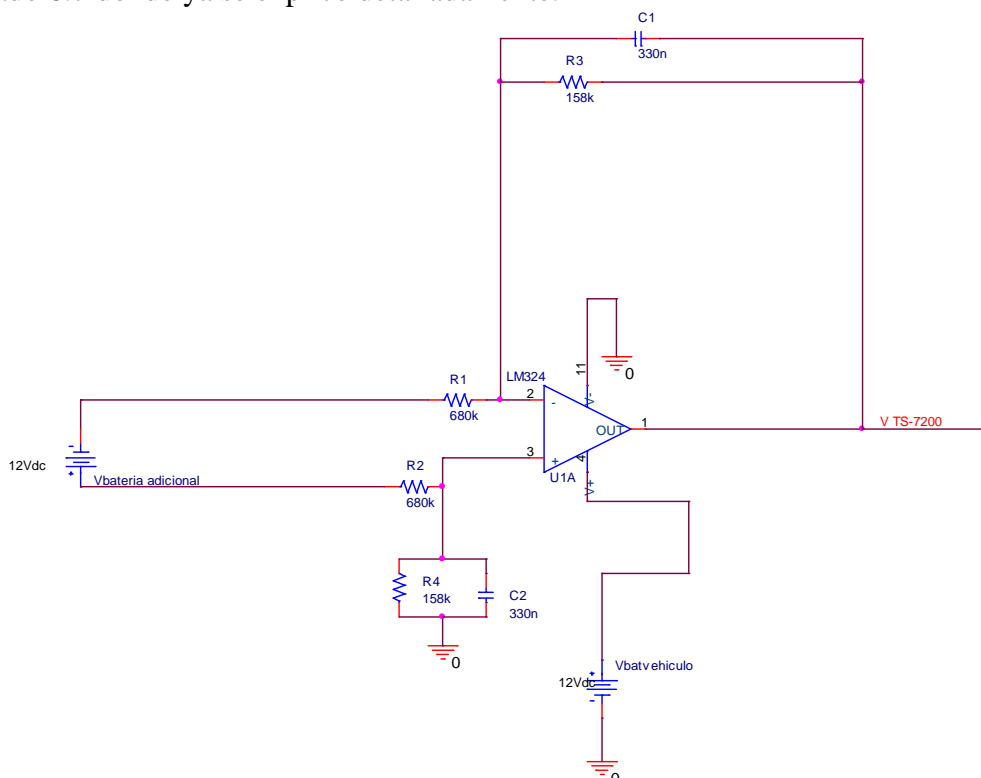


Ilustración 48: Circuito medidor de la tensión en bornes de la batería adicional

### 7.1 CIRCUITO MEDIDOR DE LA TENSIÓN EN BORNES DE LA BATERÍA

Como se demostró en el apartado 6.7.1 la ganancia de este circuito dependía del valor de las resistencias de acuerdo con la ecuación (52). Para determinar el valor de las resistencias se necesita determinar la ganancia máxima del circuito. La máxima tensión de entrada que admite la tarjeta TS-7200 son 3,3V, por seguridad para mantenerse dentro del rango de los valores de entrada se selecciona una entrada máxima a la tarjeta de 3V. La tensión máxima que puede tener entre bornes la batería adicional son aproximadamente 12,7V. Introduciendo estos valores en la ecuación (53) se obtiene una ganancia máxima de 0,23V/V. Finalmente se seleccionan

$R_3=R_4=158K\Omega$  y  $R_2=R_1=680K\Omega$  ya que son valores comerciales de resistencias que cumplen con el valor de ganancia de  $0,23V/V$  (ver ecuación (54)).

$$G = \frac{R_3}{R_1} = \frac{R_4}{R_2} \quad (52)$$

$$G_{\max} = \frac{V_{TS-7200\max}}{V_{BATERIA\_ADICIONAL\max}} = \frac{3,0V}{12,7V} = 0,23 \frac{V}{V} \quad (53)$$

$$G = \frac{R_3}{R_1} = \frac{R_4}{R_2} = \frac{158k\Omega}{680k\Omega} = 0,23 \frac{V}{V} \quad (54)$$

La Tabla 12 representa los valores de tensión obtenidos a la entrada del circuito medidor de la tensión en bornes. La Ilustración 49 muestra la linealidad de los resultados.

$V_{batería\ adicional} (V)$	$V_{TS-7200} (V)$
12,7	2,94
12,6	2,79
12,4	2,88
12,3	2,87
12,1	2,81
12,0	2,80
11,8	2,73
11,6	2,69

Tabla 12: Relación entre la tensión de entrada y la de salida del circuito medidor de la tensión en bornes

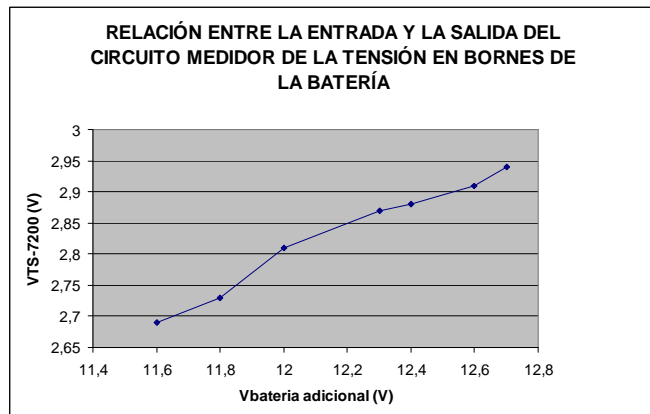


Ilustración 49: Relación entre la entrada y la salida del circuito medidor de la tensión en bornes de la batería adicional

La tensión en una batería de vehículo oscila entre  $11,6V$  y  $12,7V$  según se vio en el apartado 4.4.3. Asignado el 100% de la capacidad de la batería cuando su tensión en bornes es de  $12,7V$  y 0% cuando su tensión es de  $11,6V$  se puede establecer un rango de tensiones para determinar la capacidad de la batería (ver Tabla 13).

Tensión en bornes (V)	Capacidad aproximada
$>12,7V$	100%
$12,6V-12,4V$	75%
$12,3V-12,1V$	50%
$12,0V -11,8V$	25%
$<11,8V$	0%

Tabla 13: Rango de tensiones en bornes de una batería en relación a la capacidad de la batería



## 7.2 ACONDICIONAMIENTO A LA ENTRADA DE LA TARJETA TS-7200

Al igual que el sensor detector de red y el sensor de corriente este sensor necesita un acondicionamiento a la entrada de la tarjeta para evitar que cuando el sensor no esté funcionando, no aparezca una tensión de 1.63V proporcionada por la tarjeta si el pin está al aire. La resistencia acondicionadora será la misma que el del sensor de corriente por la similitud del circuito. El valor de la resistencia será  $4.7K\Omega$  tal y como muestra la Resistencia acondicionadora del ADC4.

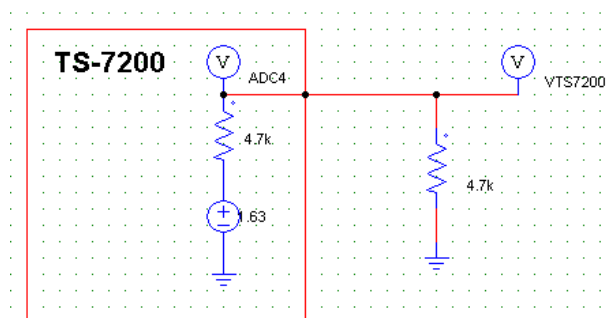


Ilustración 50: Resistencia acondicionadora del ADC4



## 8 EL SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA

### 8.1 SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA (SAI)

El sistema de alimentación interrumpida en inglés UPS *Uninterruptible Power Supply*, es un dispositivo electrónico capaz de proteger y alimentar cargas consideradas como “críticas” es decir aquellas que bajo ningún concepto puedan perder la alimentación de manera no deseada.



Ilustración 51: SAI

### 8.2 PARÁMETROS DE UN SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA

Los parámetros fundamentales para la elección son:

- Autonomía: Tiempo en el que la SAI es capaz de alimentar a la carga crítica desde el momento en el que se produce la anomalía en la red de alimentación.
- Potencia: Potencia máxima que puede tener la carga crítica.

### 8.3 TIPOS DE SISTEMAS DE ALIMENTACIÓN ININTERRUMPIDA

Existen diferentes tipos de SAIs en el mercado. Por un lado están aquellas SAIs que se utilizan en instalaciones de baja potencia y de bajo coste donde la carga es alimentada por la red y tan sólo cuando esta falla, se alimenta de la corriente alterna generada por SAI. Un problema habitual de estos equipos es que no proporcionan una senoidal pura, además de no proteger totalmente a la carga, debido a que esta no está totalmente aislada. Justo en el instante en el que se produce el fallo se produce una interrupción momentánea de la alimentación. A este tipo de SAI se les denomina SAI off-line.

Por otro lado aquellas, en las que el suministro a la carga por la red es intercalado con el suministro por la SAI. Estas proporcionan una salida de corriente alterna independiente de la red y generada a partir de corriente continua. Su uso es generalmente industrial y se denominan SAI on-line. En ellas la carga está aislada y por tanto protegida.

Además podemos encontrar unas SAI de características intermedias a las anteriores denominada SAI de línea Interactiva o in-line. La alimentación se intercala entre la red normal y la carga mediante un  *acondicionador de red*  (AVR) [49]. Este proporciona a la carga protección frente a las fluctuaciones de tensión dentro de unos márgenes. Está formado por un transformador, y triacs [50] (conmutadores de estado sólido) que aportan una respuesta rápida frente a estas variaciones evitando que la respuesta se vea afectada. Dicha conmutación se realiza en el orden de milisegundos con lo que podría decirse que no afecta a la continuidad del suministro en la carga. Además de incluye filtros y protecciones frente a sobretensiones [48].

Mientras no se produzca un fallo la carga se alimenta de la red a través del acondicionador y ante la presencia de fallo la carga se alimenta de la red alterna que genera la SAI como ya se ha expuesto, a partir de una tensión continua. Por tanto, la carga está razonablemente aislada frente a las anomalías que pudiera presentar la red de alimentación. El uso de este último tipo de SAI está generalmente destinado a aplicaciones domesticas.

La necesidad de colocación de una SAI generalmente se debe a la ponderación subjetiva de parámetros tales como: la calidad del suministro eléctrico, el nivel de seguridad deseado o los requisitos económicos de que se disponga e incluso la meteorología (tormentas, nevadas...) [48].

## 8.4 SELECCIÓN DEL SISTEMAS DE ALIMENTACIÓN ININTERRUMPIDA

### 8.4.1 Cálculo de la potencia en Vatios<sup>9</sup>

Para seleccionar un SAI de manera adecuada, no existe homogeneidad en el cálculo de la potencia aparente necesaria para la SAI aunque básicamente todos autores coinciden en el sobredimensionamiento de los cálculos. Para estimar el dimensionamiento de la SAI, en primer lugar se ha calculado la potencia en Vatios de todos aquellos elementos cuyo funcionamiento es crítico. A la hora de seleccionar los elementos críticos se debe tener en cuenta que **el SAI es un sistema de emergencia y en ningún caso es sustitutivo de la red cuando no hay suministro eléctrico**.

A este valor es necesario incrementarlo en al menos un 15%, teniendo así en cuenta el deterioro la autonomía con el tiempo. Además, el incremento permite que el SAI no trabaje por encima del 90% de la potencia activa máxima. De este modo, obtendríamos la **potencia máxima que ha de soportar el SAI**. Siguiendo estas directrices se ha realizado la Tabla 14:

---

<sup>9</sup> Nota: Los fabricantes suelen dar el dato a plena y a media carga. En los casos en los que no se dispone de alguno de ellos, se puede realizar la siguiente aproximación: La autonomía a media carga es aproximadamente el 60-70% superior al doble de autonomía a plena carga.

Unidades	Elementos a proteger				Marca	Modelo	Consumo/unidad en W	Consumo total en W
3	PCs que incluyen:				Apple	Mac Mini	110	330
	4	Cámaras conectadas al ordenador	Firefly MV	point grey				
	1	Cámara infrarroja	Indigo	Omega				
	1	GPS sistema inercial conectado a los ordenadores	Xsens	MTI-G				
							Consumo total	330

**Tabla 14: Tabla del cálculo de potencias**

Se decidió que el consumo crítico correspondía a tres ordenadores, que equivalían a una potencia activa máxima de consumo de 330W [51]. Este valor se incrementa en un 15% obteniendo la potencia activa máxima que ha de soportar el SAI ( 55).

$$\text{Potencia activa máxima que soporta el SAI} \quad 379.5W \quad (55)$$

### 8.4.2 Elección del SAI

El SAI debe ser capaz de soportar la potencia activa estimada máxima que consumen los elementos críticos, es decir 379.5W. Calculando la potencia aparente, para este valor de potencia activa, teniendo en cuenta que el factor de potencia, en entornos informáticos, se considera de 0.6 obtenemos como resultado 632.5VA ( 56). En conclusión el SAI seleccionado tendrá que admitir valores de potencia aparente superiores a 632.5VA.

$$S = V \cdot I$$

$$P = V \cdot I \cdot \cos \varphi \Rightarrow P = S \cdot \cos \varphi \Rightarrow S = \frac{P}{\cos \varphi} = \frac{379.5W}{0.6} = 632.5VA \quad (56)$$

Se compararon precios, marcas y características propias de diseño como el tamaño, el peso, el aspecto exterior y finalmente se seleccionó el modelo: APC Back-UPS ES 700VA 230V. El cual dispone de una potencia máxima configurable de 405W (superior a los 330W estimados de consumo potencia activa máxima) y una potencia aparente de 700VA (un 10% superior a la potencia aparente máxima estimada)

De entre todas los SAIs disponibles en el mercado, el modelo: APC Back-UPS ES 700VA 230V cumplía los requisitos eléctricos dimensionales y además de una buena relación calidad-precio.

### 8.4.3 Puesta en marcha

#### 8.4.3.1 Conectar la batería

Realizar las conexiones de acuerdo con la Guía de Usuario [52].

### 8.4.3.2 Conectar el equipo

El SAI dispone de tomas de corriente con la indicación “*Battery Backup-surge protection*” que permanecen activas siempre que APC Back-UPS esté encendido. Durante una sobretensión, fallo en el suministro eléctrico o cualquier otra anomalía en la red como pueden ser picos de tensión, rayos u otras perturbaciones eléctricas. Estas tomas de corriente, proporcionan potencia durante un tiempo limitado. En ellas se deben conectar los elementos de consumo de potencia críticos, en este caso, los tres Pcs. Las tomas con la indicación “*Surge Protection Only*” están provistas de una protección frente a fallos incluso si APC Back-UPS está apagado.

### 8.4.3.3 Funcionamiento

El botón ON/OFF se utiliza para encender y apagar el SAI respectivamente. Un solo pitido corto y el indicador verde de “Power ON” confirman que el APC Back-UPS está encendido y listo para proteger el sistema. La Tabla 15 contiene información de utilidad para el manejo del APC Back-UPS en ella se puede observar la relación de indicadores visuales y estados [52]. Para resolver problemas ocasionales consultar la Guía de Usuario [52].

INDICADOR VISUAL (Power ON <sup>10</sup> - verde) (Replace Battery <sup>11</sup> -rojo)	ESTADO	INDICACIÓN AUDIBLE	La alarma termina cuando:
El LED <i>Power ON</i> está encendido	<b>Power On</b> – El SAI está suministrando potencia útil a la carga	Ninguna	No dispone de alarma
El LED <i>Power ON</i> está encendido y se apaga durante un pitido	<b>On battery</b> – El SAI está suministrando potencia a las cargas conectadas a las tomas desde su batería de reserva	Produce un pitido 4 veces cada 30 segundos	El SAI regresa al modo <i>Power ON</i> o cuando se apaga
El LED <i>Power ON</i> está encendido y parpadeando.	<b>LOW battery Warning</b> – El SAI está suministrando potencia a las cargas conectadas a las tomas desde su batería de reserva y está a punto de agotarse	Produce pitidos rápidos cada medio segundo	El SAI regresa al modo normal o cuando se apaga
El LED <i>Power ON</i> y el LED <i>Replace Battery</i> están encendidos y parpadeando alternativamente	<b>Replace Battery</b> - La batería de reserva necesita ser recargada o está en el final de su vida útil y necesita ser recargada.	Tono constante	El SAI es apagado con el botón de encendido y apagado (ON/OFF)
El LED <i>Replace Battery</i> está parpadeando	<b>Battery Disconnected</b> - La batería de reserva está desconectada o en mal estado	Tono constante	El SAI es apagado con el botón de encendido y apagado (ON/OFF)
Ninguno	<b>Overload Shutdown</b> – Se ha detectado una sobre carga mientras se produce la alimentación de las tomas de corriente.	Tono constante	El SAI es apagada con el botón de encendido y apagado (ON/OFF)
Ninguno	<b>Sleep mode</b> - La batería de reserva está completamente agotada y el SAI está esperando potencia útil para volver al modo normal	Un pitido cada 4 segundos	La potencia útil es restaurada o en caso de que no sea restaurada espera 32 segundos y se apaga.

Tabla 15: Tabla de funcionamiento de la SAI

El APC Back-UPS debe recargarse durante al menos 16 horas para asegurar suficiente tiempo de ejecución. La unidad está siendo cargada siempre que está conectado a la red. Independientemente de que esté en ON o en OFF.

<sup>10</sup> Batería en correcto funcionamiento.

<sup>11</sup> Remplazar batería.

## 9 SISTEMA DE CONVERSIÓN DE LA ALIMENTACIÓN

### 9.1 INVERSOR ELÉCTRICO

El sistema de conversión de la alimentación se compone de la batería y de un inversor. Un inversor o convertidor es un equipo electrónico que convierte la corriente continua en corriente alterna adecuada para ser consumida por los equipos instalados en el vehículo. En el caso que nos ocupa transforma la corriente continua procedente de la batería en alterna con un valor de tensión de 230V y una frecuencia de 50Hz, valores necesarios en la alimentación de los equipos. El consumo en vatios de los equipos (calculado en el apartado 8.4.1) equivale a 330W.

### 9.2 SELECCIÓN DEL INVERSOR

A la hora de seleccionar un inversor se debe tener en cuenta el valor de tensión proporcionado por la batería, la tensión demanda por los equipos y el consumo total en vatios de los mismos. El modelo seleccionado fue WHS600 12 [114], el cual convierte un valor de tensión continua de 12V en tensión alterna a 230V con una frecuencia de 50Hz, para una potencia máxima de 600W.

12V =

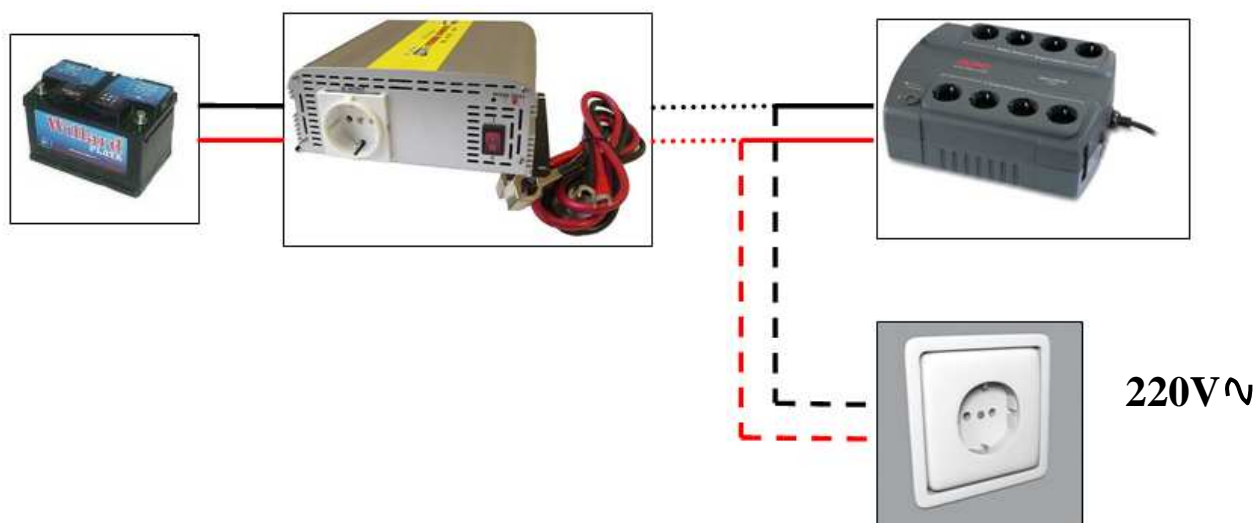


Ilustración 52: Integración del Inversor en la arquitectura

### **9.3 INTEGRACIÓN DEL INVERSOR**

La SAI podrá estar conectada directamente a la red o al inversor en función de si la tensión de red está disponible o no lo está. En el caso en el que el vehículo no esté conectado a la red, los equipos se alimentarán de la batería adicional que se ha incorporado al vehículo mediante el inversor tal y como muestra el esquema de conexiones de la Ilustración 52.



## 10 TS-7200

### 10.1 INTRODUCCIÓN A LA TARJETA

#### 10.1.1 TS-7200

La tarjeta TS-7200 es un ordenador de placa única compacto (SBC *Single Board Computer*) basado en el procesador EP9302 de Cirrus [53], desarrollado en ARM9 [54], que permite soporte de sistemas operativos tales como Linux, Windows CE, entre otros. Fabricada por Technology Systems, es una tarjeta diseñada para el desarrollo de sistemas embebidos<sup>12</sup>.

#### 10.1.2 Series y modelos

Technology Systems ofrece una amplia gama de ordenadores de placa única. Aunque muchos son parecidos no todas las tarjetas de esta serie tienen las mismas funcionalidades ni el mismo funcionamiento. Será necesario evaluar adecuadamente las necesidades y recursos que va a requerir el proyecto antes de seleccionar un determinado modelo. Por ejemplo, la placa TS-7350 no sería apta para sistemas, en los que no se disponga de una alimentación continua de la placa. Ya que, para una mayor seguridad durante el grabado de información de la tarjeta externa SD (único formato que acepta ese modelo de placa) dispone de un sistema de “**Read-Only**” llamado así porque sólo permite la lectura cuando entra en dicho estado. El sistema de *Read-Only* evita la corrupción de archivos en la tarjeta aumentando la robustez del sistema, pero obliga a una laboriosa ejecución de instrucciones para salir del estado de *Read-Only* y recuperar el control de la ejecución en la placa o la alternativa de formatear la tarjeta SD, ante una interrupción imprevista de la alimentación en la placa. Es decir, este modelo sería apto en sistemas que incorporan una batería adicional que evite los fallos imprevistos en el sistema de alimentación de la placa. La tarjeta TS-7200 no dispone del sistema de “**read-only**” lo que permite que la alimentación pueda ser interrumpida sin causar problemas en la ejecución una vez reiniciado el sistema.

#### 10.1.3 Serie TS-72XX

La serie TS-7000 corresponde a ordenadores de placa única de desarrollo en ARM<sup>13</sup> [56]. La serie TS-7200 se ejecuta en un procesador Cirrus de 200MHZ en arquitectura ARM9 de bajo consumo. Los ordenadores de placa única están disponibles en multitud de configuraciones.

<sup>12</sup> Los sistemas embebidos son sistemas de computación concebidos para la realización de un número limitado de tareas que son realizadas frecuentemente en un sistema en tiempo real. En los sistemas embebidos la mayoría de los componentes se encuentran incluidos en la placa base. En general, son sistemas programados.

<sup>13</sup> La arquitectura en ARM fue producida por la empresa Acorn Computers, corresponde a una familia de microprocesadores RISC (reduced instruction set computer), que consisten en un conjunto reducido de instrucciones con 32-bit, que facilitan la programación a los desarrolladores. Para más información consultar [56]

## 10.2 DESCRIPCIÓN

La tarjeta TS-7200 estándar (ver la Ilustración 53) se compone de tres bloques fundamentales: Interfaces, Conectores y Componentes Hardware, la Ilustración 54 muestra el esquema de bloques en una primera aproximación. La placa dispone de las siguientes características:

- Sistema operativo embebido instalado TS-Linux
- CPU ARM) a 200MHZ con MMU (unidad de gestión de memoria)
- 8MB de memoria flash que guarda la información en el arranque
- RAM de 32 MB
- 2 Puertos USB 2.0 (máxima velocidad 2Mbps/s)
- 2 Puertos serie
- Puerto Ethernet (10/100 Megabits)
- 20 entradas/salidas digitales
- Timer watchdog
- Bus de expansión PC/104
- Interfaz de bus SPI
- Interfaz matriz alfanumérica LCD
- Alimentación +5VCD 450mA
- Dimensiones (9.7 x 11.5 cm.)
- Rango operativo de temperaturas (-20° a +70°C)

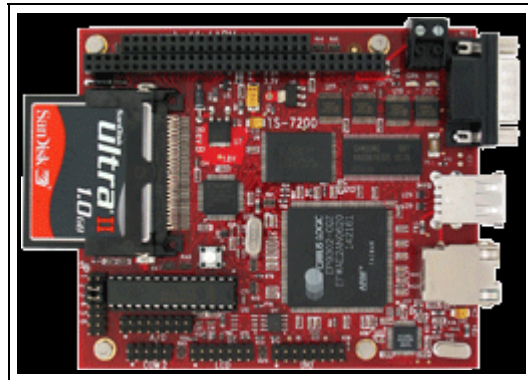


Ilustración 53: placa TS-7200

Se puede adquirir de manera adicional un kit de desarrollo que incluye cableado para la alimentación con enchufe, matriz alfanumérica, memoria flash USB entre otras opciones, para una información más detallada consultar la referencia [56].

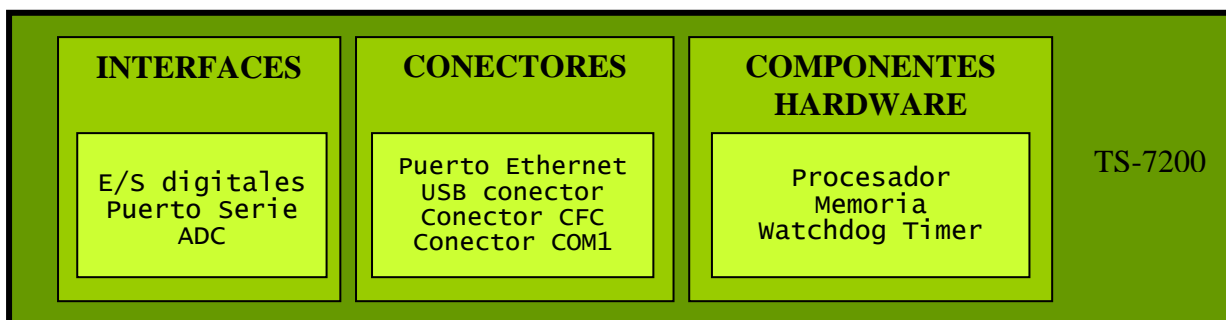


Ilustración 54: Esquema TS-7200

## **10.3 INTERFACES**

### **10.3.1 Entradas y Salidas digitales**

Existen 20 entradas/salidas digitales disponibles en la tarjeta TS-7200, repartidas en dos cabezales etiquetados como “DIO” y “LCD”. Todos las DIO son programables tanto como entradas o como salidas. Además, la dirección de cada pin puede ser individualmente programada. El cabezal etiquetado como DIO tiene 9 pines disponibles. En el pin 16 proporciona 3.3V de alimentación, limitado a una corriente de aproximadamente 1A. Parte de estos pines pueden ser programados para causar interrupciones (del DIO\_0 al DIO\_7 y del LCD\_0 al LCD\_7). Los pines DIO son individualmente configurables.

El cabezal etiquetado como LCD puede ser usado como interfaz de la matriz alfanumérica LCD. Proporciona 5V de alimentación en el pin 1 y está limitado a una corriente de 1000mA.

Aunque el procesador que incluye, es el EP9302 [57], el manual de la tarjeta TS-7200 indica que para más información se consulte el manual de usuario del modelo EP9301 [58].

### **10.3.2 Puerto Serie**

La tarjeta TS-7200 dispone de dos puertos serie asíncronos COM1 y COM2, los cuales son configurables de forma independiente. Ambos puertos tienen 16 Bytes en sistema FIFO y ambos reciben o transmiten mediante los canales de conexión UART.

### **10.3.3 Conversor A/D**

El convertidor analógico digital del EP9302 es estándar en todas las TS-7200. El EP9302 tiene cinco canales con una conversión analógico digital de 12 bit mediante un multiplexor analógico con una rango de tensiones de entrada entre 0 y 3.3V. Pero la tarjeta TS-7200 sólo tiene dos canales accesibles el ADC0 y el ADC4. Son accesibles desde el cabezal DIO1. Por tanto como máximo se puede muestrear dos valores de tensión diferentes.

Además la tarjeta TS-7200 opcionalmente dispone de 8 canales adicionales con una conversión analógico digital de 12 bit. Gracias al MAX197. Como ya se ha indicado esto es opcional, y se debe comprobar si la placa incluye este componente. Si en la placa TS-7200 está habilitada esta opción en el banner que aparece al arrancar la tarjeta aparece un texto parecido a “options MAX197-ADC” [59].

## 10.4 COMPONENTES HARDWARE

### 10.4.1 Procesador

#### 10.4.1.1 EP9302 de Cirrus

Es un procesador de 200MHZ ARM920T. El procesador contiene una unidad de gestión de la memoria MMU (*Memory Management Unity*) que permite un alto nivel de operatividad en sistemas como Linux, Windows CE o cualquiera otro sistema operativo para sistemas embebidos. En realidad el EP9302 es un procesador de propósito general al que Technologyc Systems ha adaptado los periféricos al bus estándar PC/104. El diagrama de bloques del EP9302 puede verse en la Ilustración 55. El MMU es el sistema de Gestión de memoria que lleva incorporado el microprocesador, es responsable del traslado de la dirección virtual de memoria a la física, protegiendo este proceso a través de permisos de acceso y dominios.

El núcleo ARM opera a 1.8V mientras que las entradas y salidas lo hacen a 3.3V con un consumo entre 100-750 mW. El procesador ARM920T tiene una arquitectura de 32bits, su CPU tiene 16KB de caché de instrucciones y caché de datos de 16KB. Para más información consultar el EP9301 User's Guide [58].

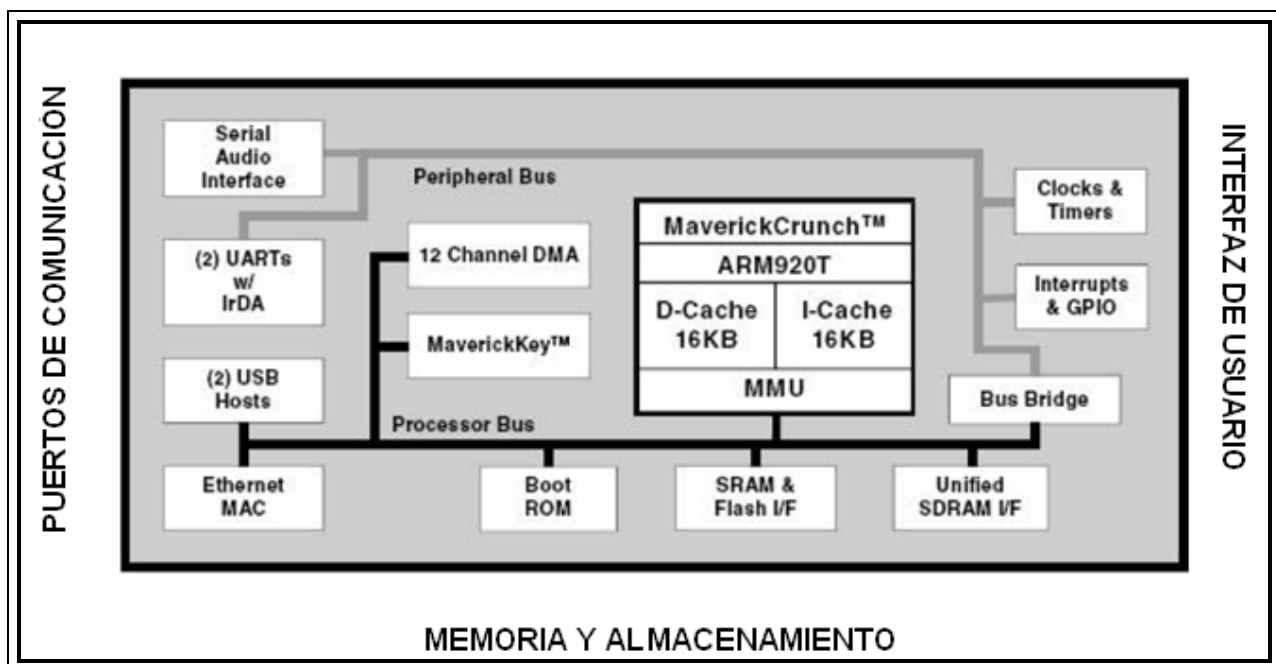


Ilustración 55: EP9302 Cirrus Diagrama de bloques

Este procesador permite controlar hasta 54 interrupciones para generar una señal de petición de interrupción IRQ (*Interrupt Request*) o de petición de interrupción prioritaria FIQ (*Fast Interrupt Request*) al núcleo del procesador. Las interrupciones internas pueden ser programadas a nivel alto o a nivel bajo. Los pines DIO programados como interrupciones pueden activarse a nivel alto, a nivel bajo, por flaco ascendente o descendente o combinándolos.

### 10.4.2 Memoria

La tarjeta TS-7200 usa tres tipos de memoria. La SDRAM es la memoria volátil de acceso rápido usada para ejecutar aplicaciones por el procesador y una memoria no volátil usada para almacenamiento de propósito general. Puede añadirse una memoria flash adicional usando los drivers de memoria USB. La tarjeta TS-7200 utiliza tecnología 32MB SDRAM, soldada directamente a la placa, para proporcionar de 32 o 64 MB de memoria volátil.

La RAM de la tarjeta TS-7200 no es continua en el mapa de memoria del EP9302 pero la MMU está programada para reasignar los bloques de RAM y aparecer como bloques continuos de memoria al comienzo del mapa virtual de memoria.

### 10.4.3 Glue Logic CPLD

La serie TS-72XX de ordenadores de placa única (SBC, *Single Board Computer*) en ARM incluyen una CPLD cuyas siglas en ingles son: *Complex Programmable Logic Device*. Una CPLD no es más que una PLD con un mayor nivel de integración. En el caso de la tarjeta TS-7200 se trata de una Xilinx 9752, la cual es responsable del control sobre los componentes de comunicación internos, mediante la implementación glue logic (familia de dispositivos del fabricante Xiling). La CPLD maneja la señal de control del bus PC104, tiene un temporizador Watchdog, controla el botón de reset y los controles de selección de la EEPROM<sup>14</sup>.

La CPLD puede ser programada mediante JTAG<sup>15</sup> *Joint Test Action Group* con el apoyo de herramientas especiales de software y hardware que proporciona Technologic Systems.

### 10.4.4 Watchdog Timer (WDT)

La tarjeta TS-7200 incorpora una unidad de WDT que puede ser usado para prevenir que el sistema se quede colgado debido a un fallo de software. El WDT causa el *reset* del sistema cuando el tiempo de espera del WDT ha finalizado, lo que permite un periodo de recuperación garantizado de un error de software. Para prevenir que actúe el WDT, la aplicación deberá realimentar continuamente al WDT.

Registro	Dirección	Acceso
Registro de control WDT	0x2380_0000	Lectura/Escritura
Registro de realimentación WDT	0x23C0_0000	Solo de escritura

Tabla 16: Registros de control del watchdog

El registro de control del WDT, cuya dirección se encuentra en la Tabla 16, puede ser inicializado con un tiempo de espera definido por el usuario, comprendido entre los 250ms y los 8s, escribiendo en el registro el valor correspondiente de acuerdo con la Tabla 17. Con la ayuda de una aplicación de software se puede reiniciar este contador en cualquier momento mediante la realimentación del WDT.

<sup>14</sup> La EEPROM, es una memoria PROM que se borra eléctricamente, puede cambiar su contenido mediante señales eléctricas y guardar contenidos cuando no se aplica alimentación, la EEPROM puede grabarse y reprogramarse un número muy elevado de veces.

<sup>15</sup> JTAG es el nombre común de la norma IEEE 1149.1 titulada Standard Test Access Port and Boundary-Scan Architecture para testear PCBs utilizando el escaneo de límites.

Valor	MSB	MID	LSB	Tiempo de Espera
0x00	0	0	0	Watchdog deshabilitado
0x01	0	0	1	250mS
0x02	0	1	0	500mS
0x03	0	1	1	1 segundo
0x04	1	0	0	Reservado
0x05	1	1	1	2 segundos
0x06	1	0	0	4 segundos
0x07	1	1	1	8 segundos

Tabla 17: Registro Watchdog para el tiempo de espera

Para la carga del registro de control WDT debe primero actualizarse el registro de realimentación y después de 30 $\eta$ S el registro de control del WDT puede ser escrito, escribir sin inicializar el registro de realimentación del WDT no produce cambios. Para poner el contador del WDT a cero se escribirá el valor 5 Hexadecimal en el registro de realimentación del WDT.

Por defecto un proceso usuario no tiene acceso a la dirección física en la que está mapeado el registro watchdog. En los sistemas operativos Linux se puede acceder al registro watchdog mediante la instrucción en C, *mmap()* mediante la cuál el sistema realiza un llamamiento a *dev/mem* que es un archivo especial para mapear áreas dentro de una dirección física de memoria en un proceso usuario que utilice el espacio de direcciones.

Es fundamental utilizar el WDT implementado por Technologic Systems en CPLD usar el incluido en el EP9302 puede causar serios problemas.

## 10.5 CABEZAL JTAG (Joint Test Action Group)

Este cabezal se usa para programar CPLD [39] usando herramientas de software y hardware especiales. También se utiliza para la configuración de jumpers. Ver apartado 10.4.3.

## 10.6 CABEZAL LCD

El puerto LCD [39] puede ser usado como interfaz con un display alfanumérico o 11 entradas/salidas digitales adicionales dispuestas de acuerdo con la Tabla 18. La alimentación de este cabezal se realiza a 5V

GND	Bias	LCD_WR	LCD_0	LCD_2	LCD_4	LCD_6
2	4	6	8	10	12	14
1	3	5	7	9	11	13
5V	LCD_RS	LCD_EN	LCD_1	LCD_3	LCD_5	LCD_7

Tabla 18: Configuración de pines del cabezal LCD

Hay que tener en cuenta que la placa incorpora, una resistencia de  $1K\Omega$ , en serie entre el EP9302 y los pines desde el LCD\_0 hasta el LCD\_7 de este cabezal que evita sobrecargas y limita la corriente de entrada al EP9302. El pin 14 sirve como un interfaz con un display LCD alfanumérico.

Cuando los pines se configuran como salidas, consumen entre 4mA y 8mA y un valor lógico entre 0 y por debajo de 3.3V. Cuando se configuran como entradas tienen un nivel de umbral TTL estándar entre 0 y 3.3V. El LCD\_EN, LCD\_RS y LCD\_WR usan un nivel lógico de tensión entre 0 y 3.3V. La dirección de memoria de los registros DR y DDR de los diferentes pines DIO del cabezal LCD se detallan en la Tabla 19.

Puerto	Pines DIO	Nº de bits	Registro	Dirección de memoria
PORT A	Del LCD_1 al LCD_7	8	DR	0x8084_0000
			DDR	0x8084_0010
PORT H	LCD_EN. LCD_RS y LCD_WR	3	DR	0x8084_0040
			DDR	0x8094_0044

Tabla 19: Mapa memoria LCD

## 10.7 PC/104 BUS EXPANSIÓN

El PC 104<sup>16</sup> es un bus que utiliza el bus ISA como portador en el sistema. Es una implementación compacta del BUS PC/AT ISA, que resulta muy eficaz en aplicaciones de sistemas embebidos. Este bus es muy habitual en la industria y muy usado en las tarjetas, por su robustez y compatibilidad con un gran número de periféricos del mercado y una abundante documentación para el desarrollador.

Nota: No todas las señales de la tarjeta TS-7200 están soportadas por este bus. Por ejemplo las señales DMA *Direct Access Memory*, aquellas que permiten el acceso directo a la memoria. Las especificaciones eléctricas [61] en el bus de expansión PC/104 son idénticas a las del bus PC ISA.

## 10.8 CONECTORES

### 10.8.1 Puerto Ethernet

La tarjeta TS-7200 dispone de un conector Ethernet que junto con el TS-linux que incluye la placa permite la comunicación directa de la tarjeta TS-7200 con cualquier equipo. Dispone además de un LED de actividad. Y una velocidad de 10/100.

<sup>16</sup> PC/104 es un estándar de ordenador embebido, que define el formato de la placa base y el bus del sistema. A diferencia de la clásica arquitectura ATX y bus PCI que son usados en la mayoría de los ordenadores personales. El PC/104 está diseñado para aplicaciones específicas, como adquisición de datos o sistemas de control industrial.



## 10.8.2 Conector USB

La tarjeta TS-7200 dispone de dos conectores USB conectados directamente al procesador EP9302, con las siguientes características:

- USB2.0 compatibles
- Open HCI Rev 1.0 compliant (host controller interface)
- Operativos como maestros en el bus
- Pueden ser usados para impresora, ratones, cámaras, teclados, etc.

TS-Kernel implementa todos los drivers necesarios para habilitar el USB. Para más información consultar *Linux for ARM on TS-72XX User's Guide* [60].

## 10.8.3 Conector para memoria SCF

Dispone de un conector para incorporar una memoria flash de tipo SCF *Compact flash card*, como la que aparece en Ilustración 56, desde la cual se permite la lectura y la escritura de archivos. La tarjeta TS-7200 debe ser reiniciada después de borrar o instalar la tarjeta CF. Con el reinicio conseguimos que el microPc arranque desde la tarjeta. El formato de la tarjeta CF debe estar en EXT2 para establecerse en Linux como el *root file system*.



Ilustración 56: conector tarjeta SCF

## 10.8.4 Conector COM1

El puerto COM1 RS232 usa un conector macho DB-9 en la tarjeta TS-7200. La Tabla 20 se ha obtenido del *TS-7200 User's Manual* [61] en ella se describen a los diferentes pines que conforman el conector COM1. La dirección del COM1 aparece en la dirección física **0x808C\_0000**.

Pin DB9	Señal	Nombre	Descripción
1	DCD	Data Carrier Detect	Detección de portadora de datos
2	RXD	Receive Data	Recibir datos
3	TXD	Transmit Data	Transmitir datos
4	DTR	Data Terminal Ready	Terminal de datos listo
5	GND	Ground	Tierra
6	DSR	Data Set Ready	Equipos de datos listo
7	RTS	Request to Send	Solicitud de envío
8	CTS	Clear to Send	Libre para envío
9	--	--	--

Tabla 20: Configuración de los pines del conector del puerto serie COM1



### 10.8.5 COM2

Este puerto tiene una configuración RS-232, la cual se ayuda de las señales TXD y RXD las cuales se usan para transmitir y recibir datos respectivamente. Al puerto COM2 se accede desde el pin 10 del cabezal etiquetado como COM2. Este cabezal no tiene una numeración convencional, esto está hecho así para que la numeración del cabezal COM2 coincida con la del conector DB9. COM2 se encuentra en la dirección física de memoria 0x808D\_0000.

TX-	NC	NC	RX-	NC
6	7	8	9	10
1	2	3	4	5
TX+	RXD	TXD	RX+	GND

Ilustración 57: Configuración de pines del cabezal del puerto serie COM2

COM2 opcionalmente puede configurarse como Full duplex o Half duplex en configuración RS-485 estableciendo el cableado como aparece en la Tabla 21. Puesto que ambas configuraciones acceden mediante el pin 10 del cabezal no podrán ser usados al mismo tiempo.

Cabezal COM2	Conector DB9	RS-232 Señal	RS-485 Half Duplex	RS-485 Full Duplex
1	1		TX+ / RX+	TX+
2	2	RXD		
3	3	TXD		
4	4			RX+
5	5	GND	GND	GND
6	6		TX- / RX-	TX-
7	7			
8	8			
9	9			RX-
10	--	--	--	--

Tabla 21: Configuración de los pines del conector del puerto serie COM2

Para más información sobre el protocolo RS-485 consultar: TS-7200 *Hardware Manual* – Technologic Systems páginas 24, 25 y 26 [39].



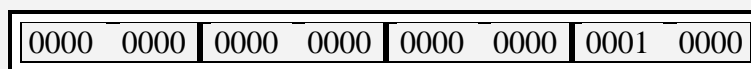
## 11 PROGRAMACIÓN DE LA TARJETA TS-7200

**E**l microprocesador trabaja internamente con números binarios, y como en cualquier dispositivo hardware es necesario manipular los bits para obtener la información o los resultados deseados.

### 11.1 REPRESENTACIÓN EN C DE LA INFORMACIÓN

C no dispone de tipo de datos específico para trabajar con números binarios, es decir con bits. Por ejemplo si se declara una variable en C de tipo entero corto se podría darle un valor entero o hexadecimal pero internamente la representación sería la misma, como se puede ver en la Figura 1.

```
short int a;  
a = 0x10;  
a = 16;
```



El microprocesador EP9302 trabaja con 32 bits. Así, el registro contiene el 16H que en binario equivale a 10000B

**Figura 1: Ejemplo representación de la Información**

Para expresar un valor binario en C se puede recurrir a las representaciones externas de tipo entero y carácter decimal, aunque la forma más sencilla de representar los números binarios en C es la representación hexadecimal.

### 11.2 MÁSCARAS

Las máscaras [61] son secuencias de bits que combinadas en operaciones binarias con otras secuencias, permite modificar estas últimas y obtener información sobre ella a conveniencia del programador. El manejo de las máscaras resulta de vital importancia ya que el EP9302 trabaja con registros modificables bit a bit, será necesario por tanto interactuar con ellos, bien forzándolos a cero, a uno o simplemente conocer el estado de un bit o grupo de bits en concreto.

### 11.2.1 Forzar bits a 0

Mediante el operador AND se puede poner a cero un bit determinado sin modificar ningún otro bit del registro ya que este operador compara los operandos bit a bit. En el ejemplo de la Figura 2, se fuerza a cero el bit 4.

En la variable *reg* se encuentra el registro al que se le realiza una operación AND (Tabla 22) con las máscara *mas* y el resultado es el mismo registro *reg* modificado. Se ha forzado a 0 el bit deseado sin alterar el resto de la configuración.

En C:

```
/*Se declara la variable*/
unsigned char reg;
reg = reg & 0x10;
```

/\*

bit AND 0 = 0  
bit AND 1 = bit

Tabla 22: Tabla AND

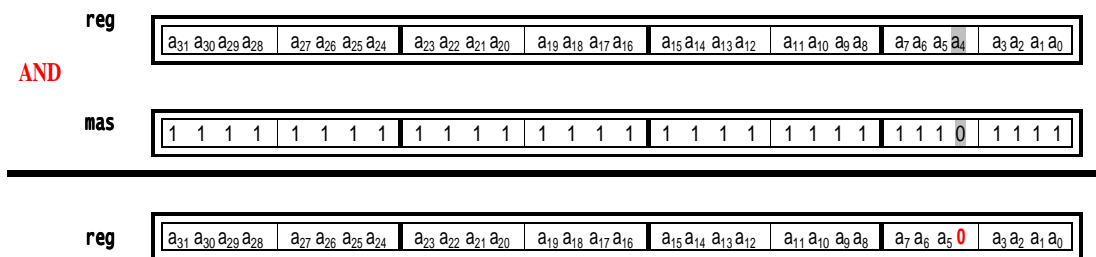


Figura 2: Ejemplo máscara para forzar bits a 0

### 11.2.2 Detectar el estado de un bit

Para conocer si un bit está a “0” o a “1”, también se usa la operación AND bit a bit. Esta vez la operación se realiza para conocer el estado del bit 5 como nuestra la Figura 3.

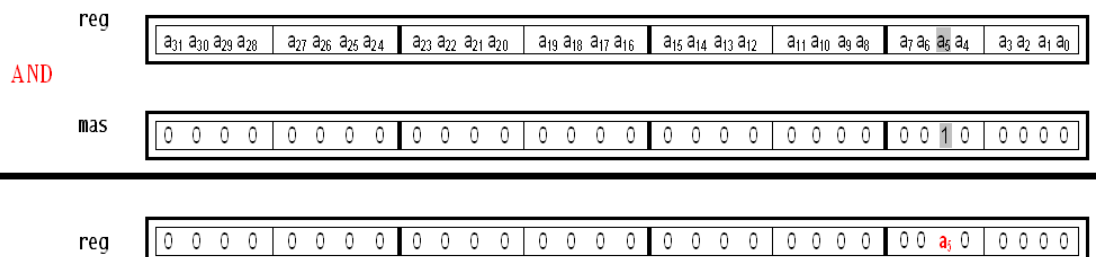


Figura 3: Ejemplo máscara para detectar el estado de un bit

Un fragmento de código C en este caso podría ser:

```
if (reg & 0x20)
    Printf("El bit 5 del registro reg está a 1");
else
    Printf("El bit 5 del registro reg está a 0");
```

### 11.2.3 Forzar bits a 1

Para forzar bits a 1 utilizamos el operador OR, para ello recordamos la tabla de funcionamiento (ver Tabla 23).

bit OR 0 = bit
bit OR 1 = bit

Tabla 23: Tabla OR

Ahora a modo de ejemplo se ponen a “1” los bit4 y 5 del registro *reg*, como se muestra en la Figura 4.

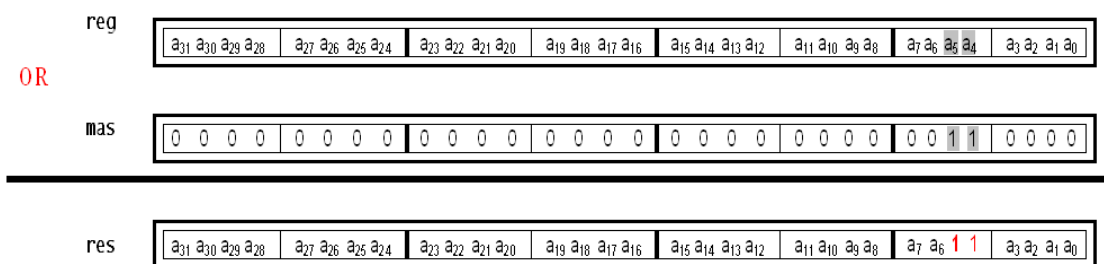


Figura 4: Ejemplo máscara para forzar bits a 1

En C:

```
/*Se declara la variable*/
unsigned char reg;
reg = reg | 0x30;
```

### 11.2.4 Complementar bits

Para complementar el estado del bit, es decir poner a nivel alto si está a nivel bajo o viceversa se usa el operador XOR, recordando la tabla de funcionamiento XOR (ver Tabla 24).

bit XOR 0 = bit
bit XOR 1 = $\overline{\text{bit}}$

Tabla 24: Tabla XOR

En este caso se han complementado los bit 4 y 5 del registro *reg* y el resto de los bits no serán modificados como se muestra en la Figura 5.

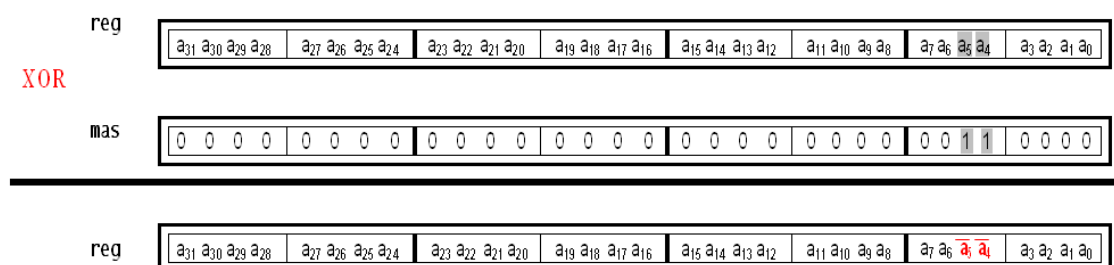


Figura 5: Ejemplo máscara para complementar bits

En C:

```
/*Se declara la variable*/
```

```
unsigned char reg;  
reg = reg ^ 0x30;
```

## 11.2.5 Desplazamientos

Los operadores `>>` y `<<` se usan para componer y descomponer registros. Un ejemplo práctico de su utilidad es el caso del MAX197 que es el conversor analógico digital opcional del que dispone la tarjeta TS-7200. El conversor es de 12 bits mientras que el registro que proporciona el EP9302 es de 32bits. También se suelen utilizar en las DIO (Digital Input Output) ya que trabajan con registros de 8 bits alineados en palabras de 32 bits.

### 11.2.5.1 Descomponer un registro

Un ejemplo sencillo se podría ver en Figura 6 donde se divide un registro de 16 bits en dos bytes.

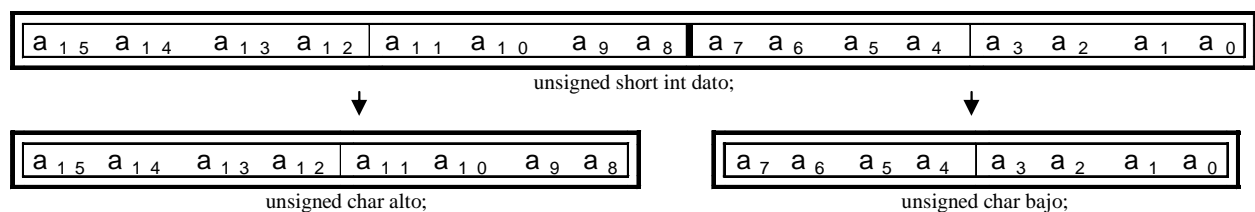


Figura 6: Ejemplo división de registro

El tipo de dato unsigned char ocupa 8 bits y el tipo de dato unsigned short int ocupa 16 bits. Es necesario tener en cuenta el espacio que ocupa cada tipo de dato para que el resultado sea el esperado. Por otro lado, *bajo* representa los bits menos significativos del registro y *alto* los bits más significativos del registro.

En C:

```
alto = dato >> 8;  
bajo = dato & 0xFF
```

### 11.2.5.2 Crear un registro a partir de otros

La forma de hacerlo sigue el mismo planteamiento del ejemplo anterior. En la Figura 7, se puede ver el ejemplo de modo análogo.

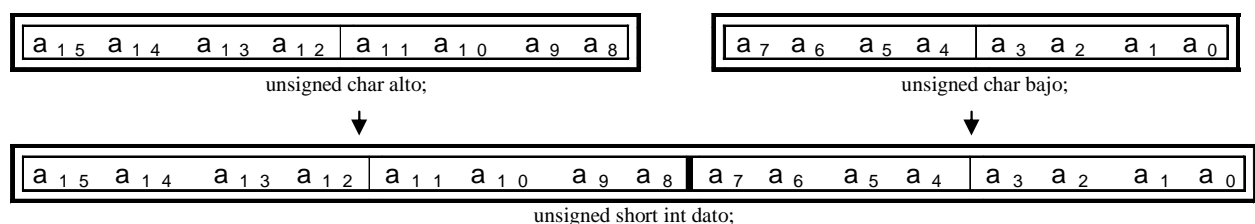


Figura 7: Ejemplo creación de un registro a partir de otros dos

En C:

```
dato = alto;  
dato = dato << 8;  
dato = dato | bajo
```

### 11.2.6 APLICACIÓN DE LAS MÁSCARAS A LA TARJETA TS-7200

La tarjeta TS-7200 proporciona señales en posiciones de memoria conocidas. El fabricante Technologic Systems proporciona un **mapa de memoria** que incluye los registros y que aparece en el apéndice de este documento. Un mapa de memoria no es más que una representación gráfica de todas las posiciones de memoria del procesador. Las máscaras se utilizan en la configuración de los pines DIO, en los LED y los conversores ADC. En realidad siempre que se trabaje con los registros de la tarjeta TS-7200.

Para indicar al sistema la dirección de trabajo mediante programación C se utiliza la función **mmap()**.

#### 11.2.6.1 *mmap(void \*start, size\_t length, int prot, int flags, int fd, off\_t offset)*

El primer argumento es una ubicación de memoria que sugiere al sistema donde debería comenzar el mapeo. El siguiente argumento es el tamaño de los bytes del archivo, en el se puede utilizar la función `getpagesize()` que devuelve el número de bytes de una página.

Al seleccionar los permisos con los que debe contar la memoria, se debe de tener en cuenta que nunca entren en conflicto con el modo de archivo. Un ejemplo de estas protecciones son `PROT_READ`, `PROT_WRITE`. Si no estamos en root no se tendrán ni permisos de lectura ni escritura sobre el archivo.

Además `mmap()` mapea los datos desde el archivo de memoria. Mediante `MAP_SHARE` se comparte con otros procesos la región mapeada. O no, mediante `MAP_PRIVATE`, con lo que conseguimos que no se vuelva a escribir en el archivo original.

El siguiente argumento es el descriptor del archivo, que habitualmente lo de nominaremos con `fd`. El valor final, será el desplazamiento que tomaremos como referencia.

Un segmento de código a modo de ejemplo sería:

```
int fd = open("/dev/mem", O_RDWR);  
assert(fd != -1);  
control = (unsigned char *)mmap(0, getpagesize(),  
                                PROT_READ|PROT_WRITE, MAP_SHARED, fd,  
                                0x10c00000);
```

## 11.3 TS-7200 y linux

### 11.3.1 Sistema operativo TS-Linux

Technologic Systems proporciona por defecto en la tarjeta TS-7200, un sistema operativo propio llamado *TS-Linux* (núcleo Linux 2.4) [94]. Este sistema operativo es una distribución de Linux mínima que incluye *Busybox*, servidor web y servidor ftp entre otras funcionalidades. El sistema *Busybox* [93] es una implementación de las utilidades más comunes estándares de Linux. La tarjeta TS-7200 dispone de la versión 2.4.26-ts1 de Linux, que se obtiene introduciendo el comando **uname -a** en la consola o terminal.

### 11.3.2 Principales comandos BUSYBOX

Comando	Utilidad
<b>ls</b>	Lista el contenido del directorio
<b>cd</b>	Permite cambiar el directorio
<b>pwd</b>	Muestra la ruta del directorio actual
<b>chmod 755 Nombre_programa</b>	Habilita al propietario los permisos de lectura escritura y ejecución
<b>./Nombre_programa</b>	Ejecuta el programa de nombre <i>Nombre_Programa</i>
<b>uname -a</b>	Obtiene la versión de Linux instalada
<b>Dpkg -l []</b>	Muestra paquetes instalados
<b>Comando --help</b>	Muestra ayuda sobre el comando

Tabla 25: Comandos Linux de utilidad

Comandos básicos de Linux como **apt-get install**, **apt-get update** o **cat/erc/issue** que se utilizan para instalar paquetes, actualizar el sistema o conocer la versión de Linux instalada respectivamente, no están disponibles en el Busybox. Para poder utilizarlos sería necesario instalar el sistema operativo Debian [95] en una memoria flash adicional.



## 12 CABEZAL DIO1

El cabezal DIO1 dispone de una salida que proporciona +3.3V, una conexión GND y 9 entradas/salidas digitales numeradas llamadas DIO (*Digital input output*) y los dos únicos canales<sup>17</sup> (accesos físicos) del ADC del procesador EP9302 que dispone la tarjeta TS-7200, a continuación en la Ilustración 58 se detalla la **configuración del cabezal DIO1**<sup>18</sup>. Donde todos los pines del cabezal DIO usan los niveles lógicos de 0 y 3.3V.

GND	Port_C0 ADC0	SPI_Frame ADC4	DIO_8	SPI_MISO	SPI_MOSI	SPI_CLOCK	+3.3V
2	4	6	8	10	12	14	16
1	3	5	7	9	11	13	15
DIO_0	DIO_1	DIO_2	DIO_3	DIO_4	DIO_5	DIO_6	DIO_7

Ilustración 58: Configuración de pines del cabezal DIO1

Todos las DIO son programables individualmente como entradas o como salidas. Los registros de control DIO son de 8 bits de anchos y alineados en palabras de 32bits. En todos los registros del EP9302 la parte superior de 24 bits no se modifican cuando se escriben y se leen siempre como ceros [61]. Cuando los pines DIO son configurados como entradas tienen un nivel de umbral TTL estándar entre 0 y 3.3V. Cuando los pines DIO son configurados como salidas el nivel lógico oscila entre 0 y 3.3V y una corriente entre 4-8mA.

Las líneas DIO (desde DIO\_0 hasta DIO3) tienen una resistencia de pull-up de 4.7K $\Omega$  a 3.3V, lo que hace es que estén a “1” si no se hace nada con ellas (trabajan en lógica negativa) los otros pines DIO tienen resistencias de polarización de 100 K $\Omega$ .

### 12.1 REGISTROS DR Y DDR

Cada entrada o salida DIO tiene dos registros que se usan para acceder a ella: Un registro de datos de 8 bits llamado **DR** (*Data Register*) y un registro de dirección de datos llamado **DDR** (*Dirección Data Register*) también de 8 bits.

DDR
Bit a “1” = salida
Bit a “0” = entrada

Ilustración 59: Registro DDR

<sup>17</sup> El ADC del EP9302 dispone de 5 canales ADC de los cuales sólo dos están accesibles en la tarjeta TS-7200.

<sup>18</sup> Esta información es especialmente útil, ya que la configuración del cabezal DIO es una información conseguida en el foro de la tarjeta TS-7200 ya que no hay información alguna en ninguno de los manuales oficiales de los que dispone la tarjeta. La única información que aparece, es una tabla con el listado de pines pero no de su posición en el cabezal.

El DDR controla si cada pin DIO es una entrada o una salida. Un bit a “1” en este registro indica una salida, mientras que un bit a “0” una entrada. La Ilustración 59 resume los estados del registro DDR.

Para conocer el estado de los pines DIO (si están como entradas o salidas) basta con leer el registro DDR. Escribir el registro de datos (DR) sólo afecta a los pines que se han configurado como salidas. Para conocer el estado de los pines DIO se realizará una lectura del registros DR.

## 12.2 INTERRUPTACIONES EN LAS DIO

Además las DIO (de DIO\_0 a DIO\_8) pueden ser configuradas para causar interrupciones. Las propiedades de interrupción de estos pines, son configurables individualmente, cada interrupción puede ser a nivel alto o a nivel bajo, por flanco ascendente o por flanco descendente.

## 12.3 MAPA DE MEMORIA DEL CABEZAL DIO1

En la Tabla 26 se detallan las direcciones de memoria de los registros DR y DDR de los pines DIO. La lectura, escritura o modificación de los registros se debe realizar de forma cíclica.

Puerto	Pines DIO	Nº de bits	Registro	Dirección de memoria
PORT B	Del DIO_0 al DIO_7	8	DR	0x8084_0004
			DDR	0x8084_0014
PORT F	DIO_8	3	DR	0x8084_0030
			DDR	0x8094_0034
PORT C	PORT_C (pin4)	1	DR	0x8094_0008
			DDR	0x8094_0018

Tabla 26: Direcciones de memoria de los pines DIO

- Las líneas DIO del DIO\_0 al DIO\_7 usan por el puerto B del EP9302.
- La DIO\_8 (pin 8) usa el bit 1 del puerto F en el EP9302.
- El pin4 del cabezal DIO1 usa por defecto el bit 0 del puerto C del EP9302.

## 12.4 GPIO (General purpose Input/Output)

Para trabajar con los Registros de datos (DR) y la dirección del registro de datos (DDR) se puede consultar el manual: *User's Guide EP9301I* en el capítulo 21 [58]. Este capítulo detalla las GPIO *General purpose Input/Output*, entradas y salidas de propósito general, de donde se ha extraído la Ilustración 60. Que relaciona las GPIO con los puertos del procesador del EP9302.

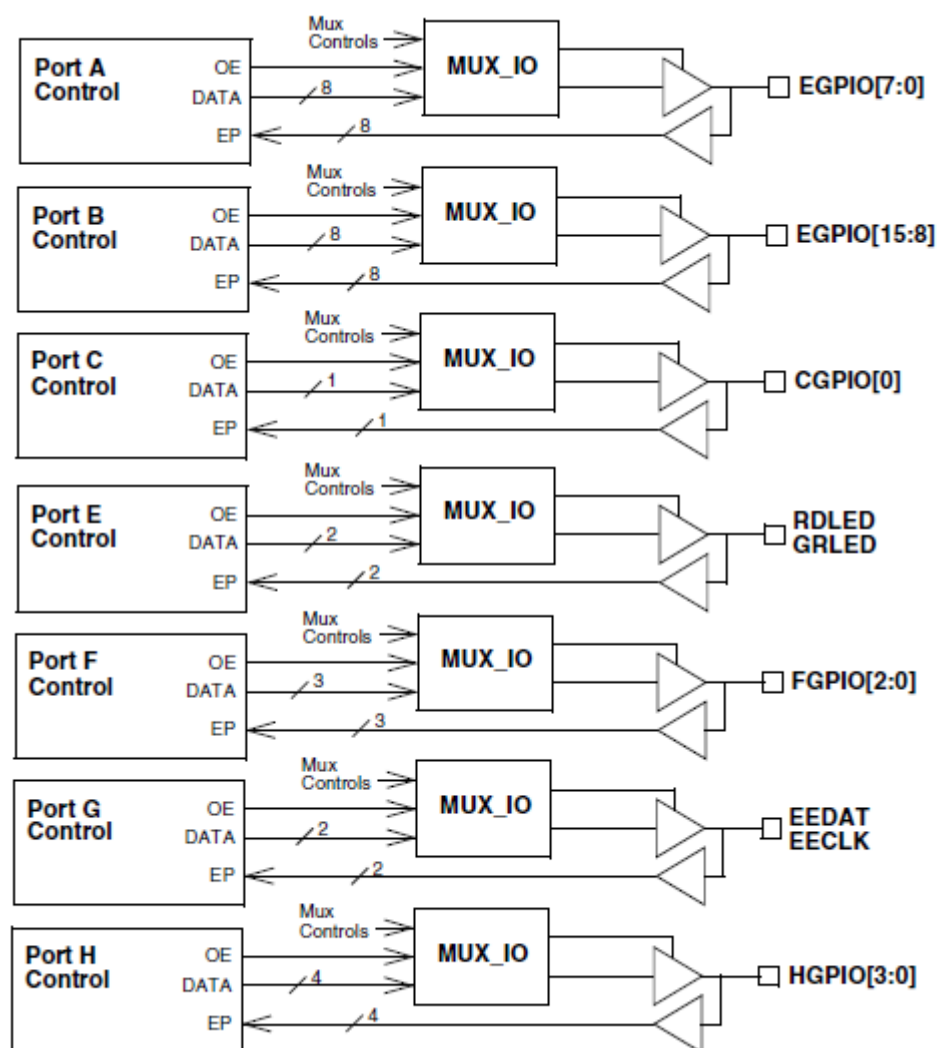


Ilustración 60: Relación entre los puertos y los GPIO

Existen dos tipos de GPIO, los GPIO estándar y los GPIO *enhaced* (mejorados), estos últimos llamados EGPIO son capaces de generar interrupciones. En la Tabla 27 se puede observar el tamaño de los puertos.

Los registros tienen un tamaño de 32bits, cada puerto tiene 8 bits DR y 8 bits DDR, el control de estos bits es individual; Es decir quedan 24 bits que no son ni accesibles ni modificables mediante escritura y se leen siempre como ceros, por esta razón se dice que son *bits reservados*.

Puerto	Tamaño
A	8 bits
B	8 bits
C	1 bits
E	2 bits
F	3 bits
G	2 bits
H	4 bits

Tabla 27: Tamaño de bits de los puertos

Los puertos A, B y F son capaces de generar interrupciones, las cuales son configurables individualmente.

Como ya se ha mencionado con anterioridad la dirección reservada a los GPIO es a partir de la dirección: 0x8084\_0000. Una vez dicho esto, queda más clara la relación entre los GPIO del EP9302, sus puertos y las DIO de la placa TS-7200.

La Tabla 28 describe siete registros adicionales para los puertos A, B y F [65]. Cada registro tiene una función y su nombre comienza por GPIO seguido del nombre del puerto y de la función que realizan.

Nombre	Registro	Función
<b>GPIO Interrupt Enable Registers</b>	GPIOAIntEn	Controlan cuales de los bits están configurados como interrupciones. Un bit a “1” indica que ese pin tiene habilitada una interrupción.
	GPIOBIntEn	
	GPIOFIntEn	
<b>GPIO Interrupt Type 1 Registers</b>	GPIOAIntType1	Determina el tipo de interrupción. Poniendo un bit a “1” en este registro se configura aumenta la prioridad de la interrupción.
	GPIOBIntType1	
	GPIOFIntType1	
<b>GPIO Interrupt Type 2 Registers</b>	GPIOAIntType2	Determina la polaridad de la interrupción. Poniendo el bit a “1” en este registro se establece que la interrupción se hará por flanco de subida y con un nivel de prioridad alto y poniendo a 0 implica que la interrupción se hará mediante un flanco de bajada o con un nivel bajo de prioridad.
	GPIOBIntType2	
	GPIOFIntType2	
<b>GPIO End-off-interrupt Registers</b>	GPIOAEOI	Se usan para desactivar las interrupciones en el Interrupt Status Register (Registro del estado de las interrupciones). Para eliminar la interrupción se pone a “1” el bit correspondiente.
	GPIOBEOI	
	GPIOFEOI	
<b>GPIO Debounced Registers</b>	GPIOADB	Elimina la habilitación de rebotes [66] que aparecen al utilizar dispositivos mecánicos de conmutación. Se utiliza en interrupciones que se producen por ejemplo mediante interruptores.
	GPIOBDB	
	GPIOFDB	
<b>Interrupt Status Registers</b>	IntStsA	Proporciona el estado de cualquier interrupción pendiente sin máscara.
	IntStsB	
	IntStsF	
<b>Raw Interrupt Status Registers</b>	RawIntStsA	Proporciona el estado de cualquier interrupción pendiente, sin importar su máscara.
	RawIntStsB	
	RawIntStsF	

Tabla 28: Registros adicionales de los puertos A, B y F

## 12.5 EL PUERTO B

El siguiente apartado, del puerto B [67], se incluye en el documento, por la importancia que supone en el manejo de las DIO en la tarjeta TS-7200. La Ilustración 61 representa los principales registros que controlan el puerto B y su implementación en el EP9302. Los registros más significativos del puerto B son el registro de datos DR y la dirección del registro de datos DDR, al pertenecer al puerto B se les denomina PBDR y PBDDR respectivamente.

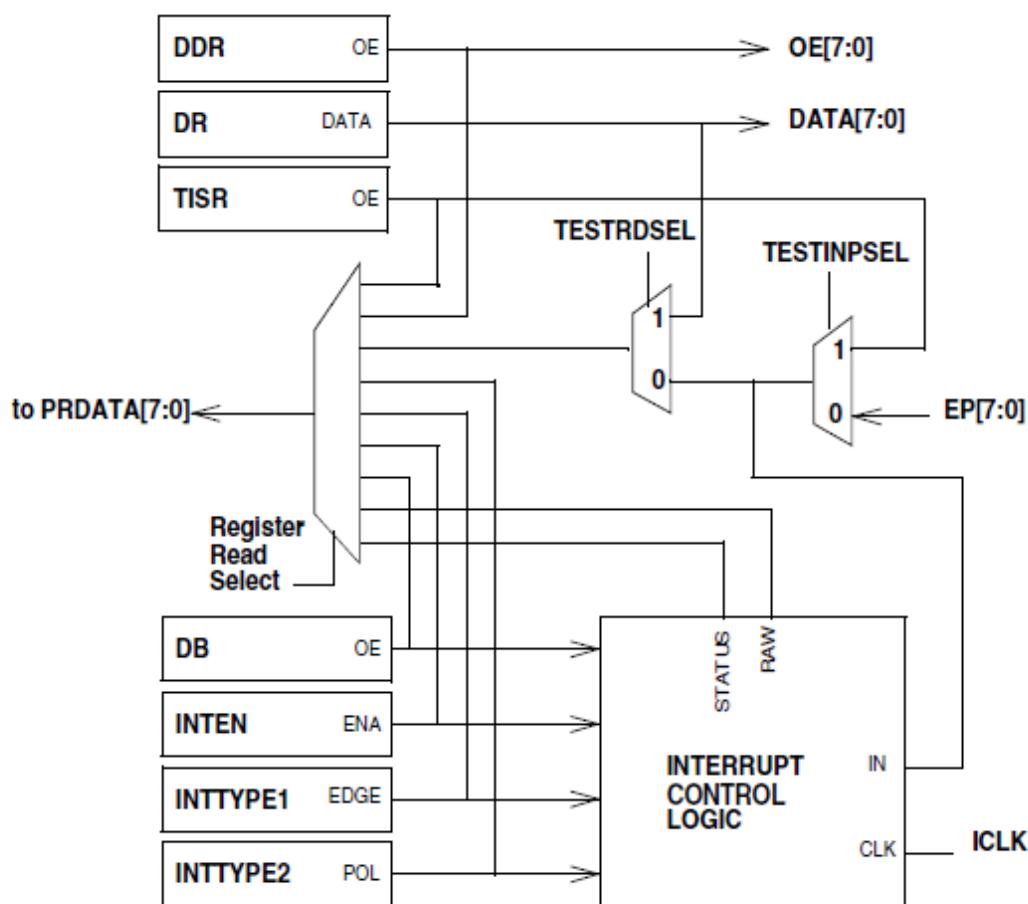


Ilustración 61: Señales de conexión de la lógica de control del puerto B

### 12.5.1 PBDDR

Al igual que todos los registros mencionados con anterioridad, el registro PBDDR dispone de los 24 bits más significativos reservados. Los 8 restantes (ver Ilustración 62) sirven para configurar como entrada o salida las DIO de la tarjeta TS-7200, de forma que la DIO\_0 le corresponde el bit 0, a la DIO\_1 le corresponde el bit 1 así hasta la DIO\_7 a la que corresponde el bit 8. La DIO\_8 está en el puerto C (ver APENDICE B: Mapa de memoria).

Un bit a nivel alto en este registro implica que la DIO correspondiente está configurada como una salida y un nivel bajo como una entrada. Hay que tener en cuenta, que un *reset* pone a nivel bajo todos los bits, es decir, los configura como salidas.

#### PBDDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PBDIR							

Ilustración 62: Registro PBDR dirección 0x8084\_0014

## 12.5.2 PBDR

Como se observa en la Ilustración 63, de nuevo, los 24 bits más significativos están reservados. A la hora manejar los registros es fundamental tener en cuenta que:

- ✓ Los valores escritos en los bits, aparecerán en la salida DIO de la tarjeta TS-7200, siempre que ese mismo bit esté a nivel alto en el registro PBDDR, es decir cuando estén configurados como salidas.
- ✓ Los valores leídos en este registro reflejan el estado de la DIO configurada como entrada, siempre que el mismo bit esté a nivel bajo en el registro PBDDR.

Se debe recordar que los valores de escritura solo afectan a los bits configurados como salidas. Los dos puntos anteriores son la clave del funcionamiento de las DIO.

### PBDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PBDATA							

Ilustración 63: Registro PBDR dirección 0x8084\_0004 con posibilidad de lectura/escritura

## 12.6 INTERFAZ SPI

El puerto serie síncrono, del EP9302 está disponible en el cabezal DIO, este puede ser implementado tanto como esclavo o como maestro, para dispositivos periféricos con interfaces de *Motorola SPI* [68] o *National Microwire serial* [69], en los cuales el *path* de la transmisión y recepción de datos es almacenado mediante un sistema FIFO en la memoria.

Para usar algunos de los pines DIO, hay disponible una interfaz para direccionar hasta 9 periféricos [70]. Los pines del bus SPI están definidos en la Tabla 29 que aparece a continuación. Dichos pines funcionan entre los niveles lógicos entre 0 y 3.3V.

Pines DIO1	Nombre	Función
10	SPI_MISO	Maestro entrada / Esclavo salida
12	SPI_MOSI	Maestro salida / Esclavo entrada
14	SPI_CLK	Reloj
6	SPI_Frame	SPI Frame Pin

Tabla 29: Pines del cabezal DIO1 correspondiente a los SPI

## 12.7 MATRIZ DE TECLADO

Las señales DIO están situadas de tal forma que permiten la conexión de un teclado de matriz de 4x4 conectado desde la DIO\_0 hasta la DIO\_7.

## 13 CONVERSOR ANALÓGICO DIGITAL

Un conversor analógico digital ADC (*Analog to Digital Converter*) convierte una señal de entrada analógica, en un valor binario, es decir en una señal digital. El proceso de digitalización de una señal consta de cuatro fases:

- 1) **Muestreo:** (“*sampling*”), consiste en tomar muestras periódicas con una frecuencia fija (Hz).
- 2) **Retención:** (“*hold*”), las muestras se retienen para poder cuantificarlas.
- 3) **Cuantificación:** Se trata de medir el voltaje de cada una de las muestras.
- 4) **Codificación:** Consiste en traducir los valores obtenidos a código binario.

La placa TS-7200 dispone del conversor analógico digital EP9302 y opcionalmente puede llevar incluido el conversor MAX197 [83]. Para comprobar si la placa que se ha adquirido dispone del conversor MAX197 opcional, véase el apartado: 10.3.3. Este conversor es muy sencillo de utilizar, se dispone de un ejemplo en las referencias [84] [85]. Su utilización requiere el uso de máscaras (ver 11.2 y 11.2.5). Puesto que el conversor MAX197 no estaba incluido en la tarjeta TS-7200 de la que se disponía no se entrará en más detalles sobre él.

### 13.1 CARACTERÍSTICAS DEL CONVERSOR A/D EP9302 DE CIRRUS

Como ya se ha mencionado en varias ocasiones el EP9302 dispone de 5 canales para la conversión analógica-digital de 12bits aunque en la tarjeta TS-7200 sólo están accesibles dos de ellos: el ADC0 y el ADC4. Su rango de tensión de entrada se encuentra entre 0 y 3.3V, es decir será apto para señales de entrada que estén entre 0 y 3.3V. Puede tomar hasta un máximo de 925 muestras por segundo, requiriendo un tiempo de establecimiento de 2mS para adaptarse al cambio de canal.

La resolución del dispositivo se puede calcular según la ecuación ( 57). Esto quiere decir que por cada 12,9mV que aumente el nivel de tensión en la entrada, el conversor aumentará su salida en una unidad.

$$\text{resolución} = \frac{\text{Máx valor análogo}}{2^8} = \frac{3,3V}{2^8} = 0.01289V = 12,89mV \quad (57)$$



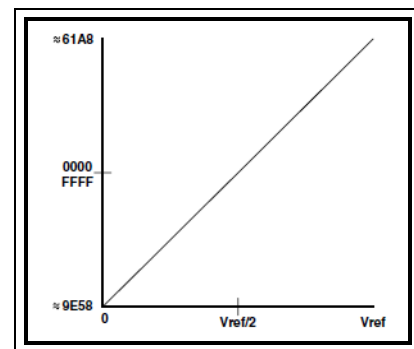
## 13.2 INTERPRETACIÓN DEL CONVERTIDOR ANALÓGICO DIGITAL A EP9302 DE CIRRUS

Cada canal del conversor A/D en la tarjeta TS-7200 está calibrado, y sus valores son almacenados en la memoria no volátil de la EEPROM. Esta calibración minimiza tanto el error de ganancia como el error de Offset en el procesador EP9302. El manual [39] aconseja utilizar los valores de 0V y 2.5V como valores de referencia [89] es importante usar los valores de acuerdo con los ejemplos que proporciona el fabricante [90] ya que los valores de calibración están almacenados en la EEPROM. Se almacenan en forma de array del tipo: [número de canal] [punto de referencia 0V, punto de referencia 5V]

Los puntos de referencia son almacenados como valores de 16 bits y se usan para corregir los valores que devuelve el conversor.

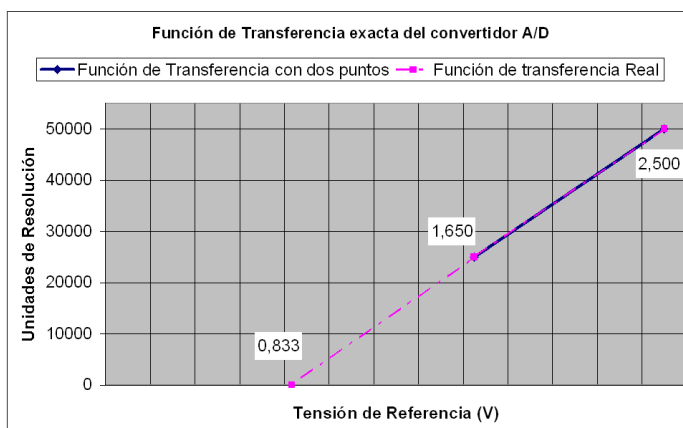
La única forma de saber si la lectura es correcta, es si el instrumento está calibrado mediante un patrón de referencia conocido. La calibración afecta fundamentalmente a la repetibilidad de la medida.

La Ilustración 64 muestra la función de transferencia del conversor analógico-digital del procesador EP9302 donde el eje de abscisas representa el valor de la tensión de referencia y en el eje de ordenadas los valores de salida del conversor.



**Ilustración 64: Función de transferencia del convertidor A/D**

Hay 25.000 valores comprendidos entre 0000H y 61A8 y otros 25.000 entre 9E58 y 61A8, lo que equivale a una resolución de 50.000 unidades, en un rango de tensión de entrada de 0 a 3,3V. Los valores de calibración se almacenan en la memoria EEPROM desde el byte 0x07EB al 0x07FE. Esto se refleja en el programa de gestión de la placa TS-7200, en la variable CALIB\_LOC de valor 2027d (equivale a 0x07FE en hexadecimal) que determina la localización de los valores de calibración en la memoria.



**Ilustración 65: Función de transferencia exacta del convertidor**

En realidad la Ilustración 64 es una gráfica simplificada de la función de transferencia real que se puede ver en la Ilustración 65. En esta última gráfica se observa que para la máxima resolución se ha tomado el valor de tensión referencia de 2.5V de acuerdo con la Ilustración 64 el valor de  $V_{ref}/2$  sería 1.65V. Alargando la recta creada por estos dos puntos se observa que cuando la resolución es cero, el valor de tensión de referencia es de 0.833V.

Los valores clave a la hora de realizar el programa que gestiona el ADC, en la placa TS-7200, son 2.5V y 0.833V. El valor de 2,5V viene dado por el fabricante de la tarjeta TS-7200.



Para realizar el proceso de conversión el ADC dispone de una máquina de estado. La conversión se inicia con la lectura por parte de la CPU del registro “*TSXYResult register*”, el contenido del registro se almacena en una variable local. La lectura del contenido continúa realizándose reiterativamente hasta que el bit SDR (*Synchronous Data Ready*, bit 31 del registro *ADCResult*) esté a “1”, este proceso puede llevar algún tiempo hasta alcanzar el número de muestras deseadas. Cuando el registro *TSXYResult* es leído, el bit SDR toma valor “0”. Después de utilizar la variable local debe aplicársele la máscara 0xFFFF que elimina el contenido de la misma. Los doce bits menos significativos del *ADCResult* (en la dirección 0x8090\_0008) contienen el resultado de la conversión.

La librería **Librería EP93XX\_adc.h**, es la librería que recoge las funciones para inicializar el ADC (función *init\_adc*) y leer en el canal deseado (función *read\_channel*).

### 13.3 CONFIGURACIÓN SOFTWARE

Para trabajar con el conversor analógico digital del EP9302 es necesario realizar una serie de configuraciones mediante software. A continuación se detallan los pasos a realizar:

#### a) Desbloquear software

El registro *ADCSWLock* situado en la dirección de memoria 0x8090\_0020 del EP9302, es el que permite realizar el desbloqueo por software. Al conjunto de los 8 bits menos significativos que aparecen recuadrados en la Ilustración 66 se les denomina *SWLCK*. Escribiendo 0xAA en *SWLCK*, se desbloquean todos los registros bloqueados hasta el siguiente bloque de acceso, es decir cada vez que se desee acceder a un registro se debe desbloquear el software previamente. En el TS-7200 Hardware Manual aparece con la dirección 0x8093\_00C0 que es en realidad la dirección del registro *SysSWLock* como aparece en *EP9301 User's Manual* [61]

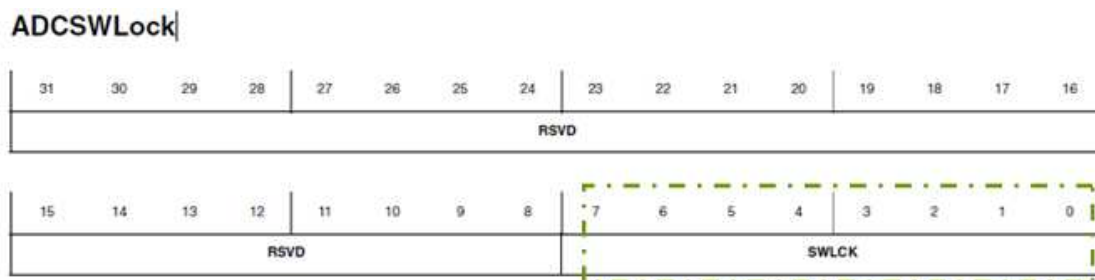


Ilustración 66: Registro *ADCSWLock* en la dirección 0x8990\_0020

#### b) Activar el bit 31 del registro *ADCClkDiv*

Forzando a “1” el bit 31<sup>19</sup> del registro *ADCClkDiv* que aparece recuadrado en la Ilustración 67, se habilitan los relojes del ADC del procesador EP9301. Se recuerda que para

<sup>19</sup> La documentación de la tarjeta TS-7200 contiene a menudo erratas o dificultades para la comprensión de sus manuales y esta es una muestra de ellas. Tanto el *EP9301 User's Manual* como el *TS-7200 Hardware manual* coinciden en que el bit 31 le llama TSEN, en vez de ADCEN como aparece en la ilustración es posible que la ilustración esté equivocada, en cualquier caso se trata del bit 31 (la ilustración proviene del *EP9301 User's Manual*).

forzar bits a “1” es necesario utilizar una máscara adecuada mediante una operación “OR”, ver apartado 11.2.3 para más información.

#### ADCClkDiv

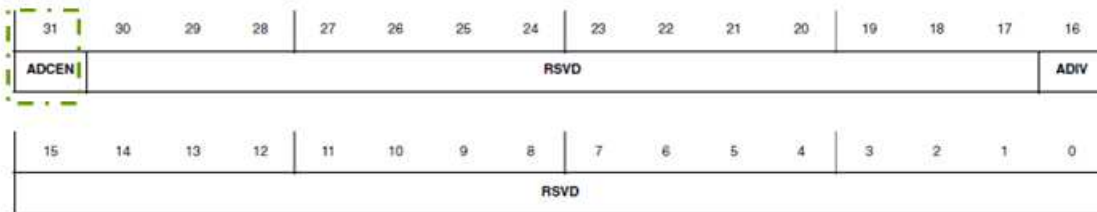


Ilustración 67: Registro ADCClkDiv mapa de memoria 0x8093\_0090

#### c) Desbloquear software

Antes de activar el bit *ADCEN* (apartado 12.2 d)) es necesario desbloquear los registros bloqueados, repitiendo el apartado 12.2 a), dado que desbloqueo por software solo permanece desbloqueado durante el siguiente ciclo.

#### d) Poner a “1” el bit *ADCEN* (*ADC Enable*)

El registro *DeviceCfg* (*Device Configuration Register*) controla las funciones principales del sistema. Forzando a “1” el bit *ADCEN*, bit 17 en el registro *DeviceCfg* [62] que aparece recuadrado en la Ilustración 68 se habilita el ADC. La guía *TS-7200 Hardware manual* [39] confunde este registro con el *ADCClkDiv*.

#### DeviceCfg



Ilustración 68: Registro DeviceCfg en la dirección de memoria 0x8093\_0080

#### e) Forzar a “0” el bit *ADCP* (*Power Down*)

En el registro *DeviceCfg* se debe forzar a “0” el bit 2, llamado *ADCPD*, con el que se activa el conversor y la señal de reloj propia del ADC. El bit *ADCPD* debe estar a cero para un normal funcionamiento del ADC.

Habilitar el *ADCEN* (apartado 12.2 d)) no afecta al estado de activación del ADC, ya que este bit, está directamente controlado por el bit *ADCPD*.

#### f) Desbloquear software

Como cada vez, que se desea acceder a un registro, es necesario desbloquear previamente, los registros bloqueados mediante software, repitiendo el apartado 12.2 a).

### g) Selección de canal en el registro ADCSwitch (Analog Switch Control registers)

El registro *ADC Switch* que aparece en la Ilustración 69 es un registro situado en la dirección: 0x8090018, dependiendo de si se utiliza un canal u otro se debe asignar un valor predeterminado [87] a este registro, que viene dado por la Tabla 30. Donde el ADC0 y al ADC4 corresponden a los pines DIO 4 y 6 respectivamente.

Entrada	Valor del Registro ADC Switch
ADC0	0x0000_0608
ADC4	0x0000_0610

Tabla 30: Valores de selección del canal del ADC

#### ADCSwitch

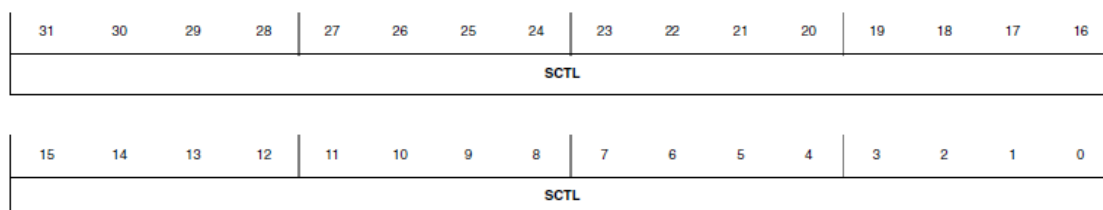


Ilustración 69: Registro ADCSwitch en la dirección 0x8090\_0018

### h) Como consultar el resultado

Como ya se explicó el apartado 13.2, se debe consultar reiterativamente el registro *TSXYResult* hasta detectar un cambio en el valor del bit SDR (que SDR = “0”) que aparece remarcado en la Ilustración 69. En esta figura puede verse también remarcados el grupo de 12 bits menos significativos denominado AD donde aparecerá el resultado de la conversión analógica digital con una resolución de 12bits.

#### ADCResult



Ilustración 70: Registro ADCResult 0x8090\_0008

## 13.4 LEDs JUMPERS Y BOTONES

### 13.4.1 LEDs

La placa TS-7200 dispone de dos LEDs, uno verde y otro rojo configurables mediante software, que pueden usarse para diagnostico, avisos, o simplemente como salidas. Al encender la placa, estos LEDs son usados por la alimentación como indicadores de que todo va bien, pero una vez arrancada la placa se puede retomar el control por software. La dirección de memoria

0x8084\_0020 corresponde al registro *PEDR Port E Data Register* (ver Ilustración 71) donde el bit 0 y el bit 1 están configurados para el control del led verde y el rojo respectivamente. Un “1” lógico en el bit correspondiente indica que el led está encendido y un “0” lógico que está apagado tal y como aparece en la Tabla 31.

## PEDR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSVD															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														PEDATA	

Ilustración 71: Registro PEDR 0x8084\_0020

Salida	Dirección de memoria	Nº de Bit	Estado del Bit	Función
LED ROJO	0x8084_0020	Bit 1	“0”	Led rojo apagado
			“1”	Led rojo encendido
LED VERDE	0x8084_0020	Bit 0	“0”	Led verde apagado
			“1”	Led verde encendido

Tabla 31: Mapa de memoria LEDs

## 13.4.2 Jumpers

La tarjeta TS-7200 contiene 6 jumpers configurados por defecto cuya descripción se detalla en la Tabla 32 que aparece a continuación. La dirección física, y el número de bits que corresponde a cada jumper puede verse en la Tabla 33. Los jumpers se activan cuando aparece un “1” lógico en el bit correspondiente a cada Jumper y se desactivan si aparece un “0” lógico.

JUMPER	Descripción
JP1	Arranque de la tarjeta TS-7200 usando el puerto serie COM1
JP2	Habilita el puerto serie COM1 como acceso a la consola
JP3	Habilita la escritura en la memoria flash
JP4	Habilita el puerto serie COM2 como acceso a la consola
JP5	Test de jumpers
JP6	Jumper disponible para el usuario

Tabla 32: Listado de jumpers

JUMPER	Dirección física	Nº de Bit	Estado del Bit
JP2	0x1080_0000	0	“1” activa el jumper JP2 “0” desactiva el jumper JP2
JP3	0x1080_0000	1	“1” activa el jumper JP3 “0” desactiva el jumper JP3
JP4	0x1080_0000	3	“1” activa el jumper JP4 “0” desactiva el jumper JP4
JP5	0x1080_0000	4	“1” activa el jumper JP5 “0” desactiva el Jumper JP5
JP6	0x220_0000	0	“1” activa el jumper JP6 “0” desactiva el jumper JP6

Tabla 33: Estado de los jumpers

### 13.4.3 Botones

La tarjeta TS-7200 dispone de un botón que aparece recuadrado en la Ilustración 72. Pulsar el botón provoca el restablecimiento completo del sistema, de igual modo que si se suprimiese la alimentación.

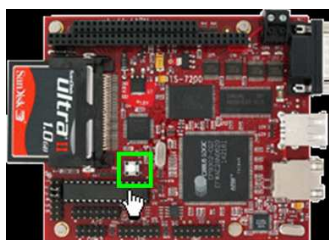


Ilustración 72: Botón de reset en la placa TS-7200



## 14 PUESTA EN MARCHA DE LA TARJETA

### 14.1 ALIMENTACIÓN DE LA TARJETA TS-7200

La placa TS-7200 requiere una tensión continua de alimentación de 5VDC a 450mA. Cuando la tarjeta está conectada al ordenador mediante Ethernet y se estabiliza consume una corriente de 320mA.

Para conectar la placa se deben seguir los siguientes pasos:

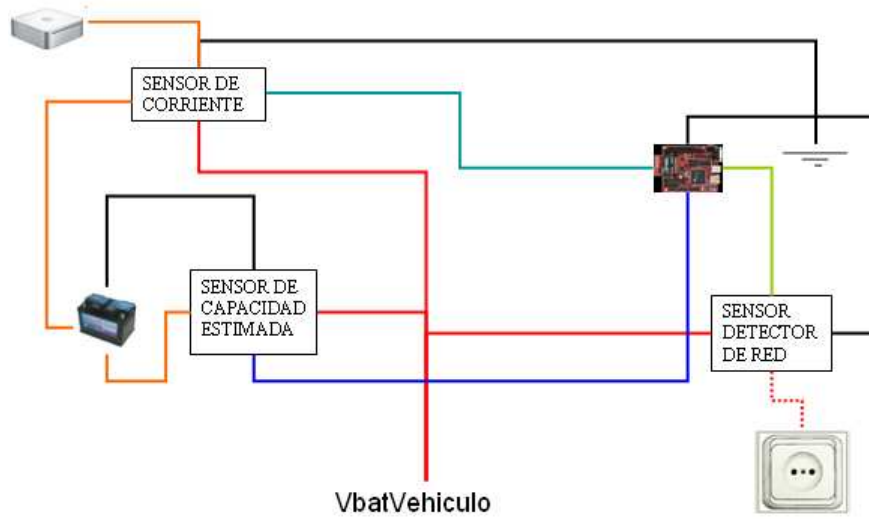
1. Se procederá en primer lugar a alimentar la placa con un voltaje de unos 5VDC. Es importante que la alimentación no superare los 6V y que tampoco sea inferior a 4V. Si la tensión cae por debajo del mínimo admisible en la tarjeta se encenderían el led verde y el rojo, que hay junto la alimentación, simultáneamente.
2. Conectar la placa al PC mediante el cable de Ethernet.
3. Conectarse vía Telnet mediante la consola Eclipse [91], el programa Putty [92] o cualquier otro de características similares.

### 14.2 CONEXIONES

El esquema de conexiones del sistema de gestión de la batería se muestra en la Ilustración 73 donde las líneas continuas representan conexiones permanentes y las discontinuas conexiones no permanentes. La Tabla 34 muestra cuales de estas conexiones son entradas o salidas de los sensores.

SENSOR DETECTOR DE RED		SENSOR DE CORRIENTE		SENSOR CAPACIDAD ESTIMADA	
ENTRADAS	SALIDA	ENTRADAS	SALIDA	ENTRADAS	SALIDA
Vred	DIO_0 TS-7200	Iconsumida por los equipos	ADC0 TS-7200	V batería adicional	ADC4 TS-7200
Vbatería del vehículo		Vbatería del vehículo		Vbatería del vehículo	
GND		GND		GND	

Tabla 34: Relación de entradas y salidas



**Ilustración 73: Esquemático de conexiones**



## 15 ENTORNOS DE DESARROLLO

La tarjeta TS-7200 no dispone de controlador de vídeo ni de interfaz a través del teclado, por eso, se necesitan diferentes entornos desde que se genera el código hasta que se ejecuta en la tarjeta. Mediante el programa Eclipse se compila el código en lenguaje C y se crea un ejecutable que es transferido a la memoria de la tarjeta, mediante el programa FileZilla [96]. Para ejecutar el código y comunicarse con la tarjeta se usa el programa Putty, la herramienta *Console* en Eclipse u otro programa de características similares. En resumen, los entornos utilizados son:

<i>Eclipse</i>	→	Se utiliza para obtener un ejecutable para la arquitectura de la tarjeta TS-7200
<i>FileZilla</i>	→	Su función es transferir el ejecutable mediante el protocolo FTP (sustituye a la herramienta <i>Remote System</i> de Eclipse)
<i>Putty</i>	→	Establece comunicación con la tarjeta mediante el protocolo de red Telnet (puede ser sustituido por la herramienta <i>Terminal</i> en Eclipse).

### 15.1 ECLIPSE

Eclipse es un entorno de desarrollo para diferentes lenguajes (java, c, c++, e incluso Python [105] si se configura adecuadamente) bastante completo pero no es fácil para un usuario novel. El programa Eclipse cuyo logotipo puede verse en la Ilustración 74, se obtiene de forma gratuita en su página oficial [106]. El desarrollo de la programación se ha realizado con el paquete Eclipse Europa.



Ilustración 74: Logotipo Eclipse

El primer paso para entender el funcionamiento de Eclipse es comprender, que Eclipse está dirigido a un sistema operativo Linux, esto quiere decir, que en Windows funcionará siempre y cuando se adapte adecuadamente, aunque su potencial se reducirá [107].

La versión de Eclipse que proporciona el fabricante para la placa TS-7200 necesita un sistema operativo Debian en el PC en el que se instala Eclipse y un sistema operativo Debian en la placa TS-7200. La tarjeta adquirida no dispone de espacio suficiente en la memoria, como para instalar el sistema operativo Debian. Para instalarlo sería necesario aumentar la memoria, bien mediante una memoria flash USB o bien mediante una tarjeta CF.

Otra manera de utilizar Eclipse es usarlo a modo de compilador y una vez que se obtiene el ejecutable, transferirlo a la placa mediante protocolo FTP. Almacenando en la tarjeta TS-7200 únicamente los ejecutables, se puede evitar utilizar una memoria adicional. Esta será la solución finalmente adoptada.

### 15.1.1 Entorno eclipse. configuración de la ventana de trabajo

La ventana principal que aparece al abrir el entorno de desarrollo Eclipse es la que aparece en la Ilustración 75.

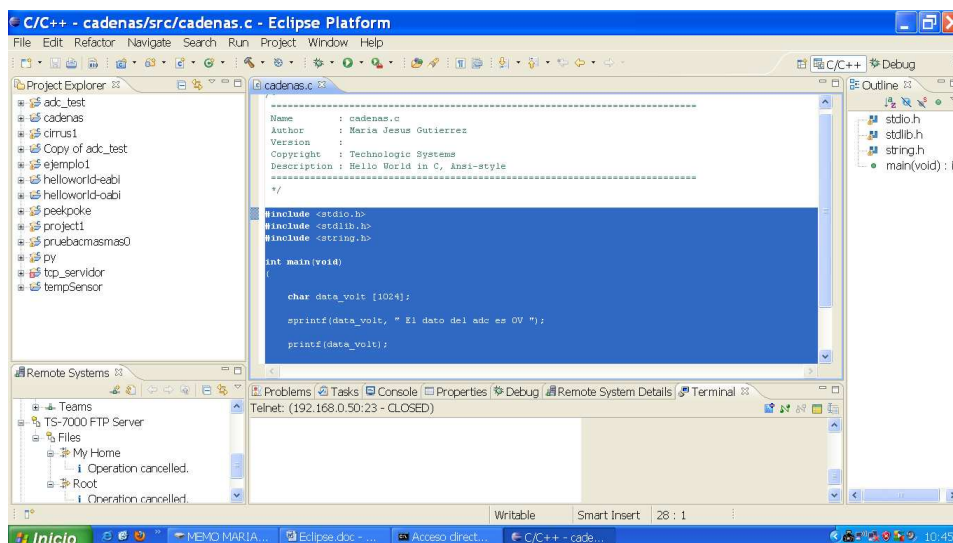


Ilustración 75: Ventana de trabajo Eclipse

En el caso de que al abrir por primera vez Eclipse no estuviera configurada la ventana de trabajo conforme a la Ilustración 75 es aconsejable reconfigurarlo. Para configurar la pantalla de trabajo en primer lugar se abre la perspectiva de trabajo *C/C++ (default)* para ello se hace click en **Window** situado en el menú principal: **Window**→**Open Perspective**→**Other** como se muestra en la Ilustración 76. En **Other** se selecciona la opción *C/C++ (default)*.

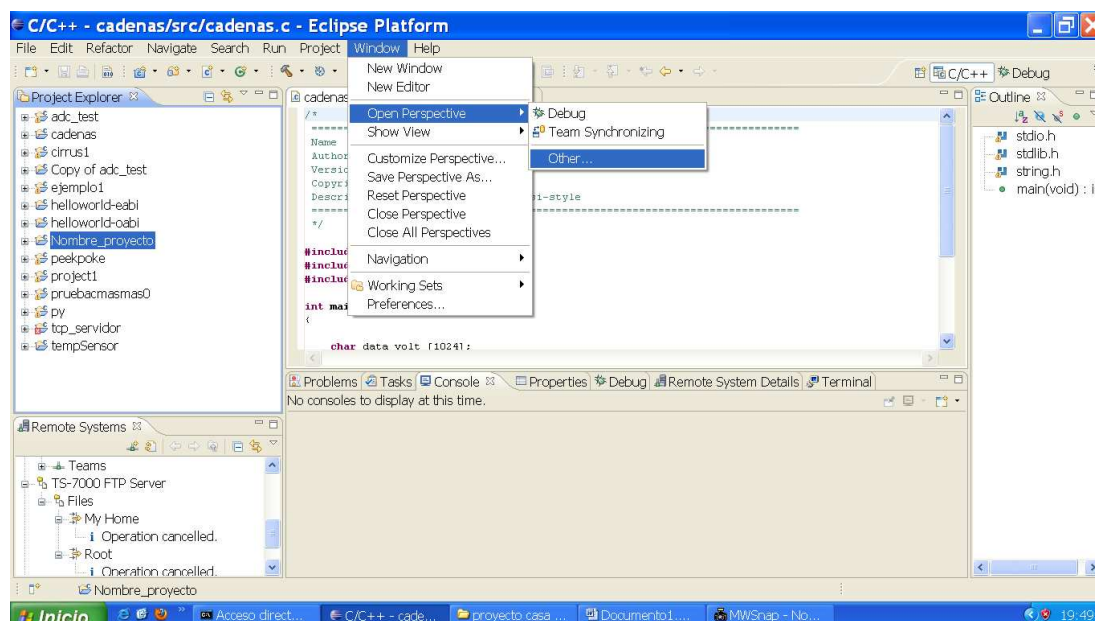


Ilustración 76: Configurar pantalla

Las vistas son pestañas arrastrables, situando el cursor sobre la pestañas pueden ser arrastradas hasta configurar la ventana de trabajo según la Ilustración 75. Si faltase alguna vista se puede añadir haciendo click en menú Windows de la barra de menú principal, **Show View→Other→Terminal** y **OK**. La opción **Show View** muestra todas las vistas disponibles. En la Ilustración 77 se ha añadido a modo de ejemplo la vista: **MakeTargets** Para eliminar una vista de la ventana de trabajo bastará con cerrarla.

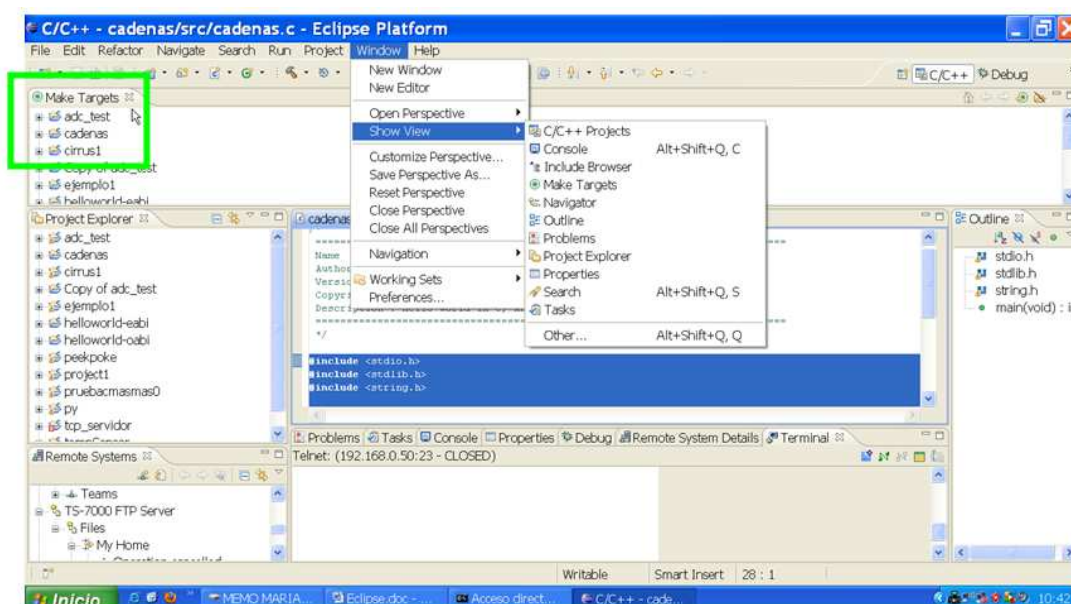


Ilustración 77: Configuración de la ventana de trabajo

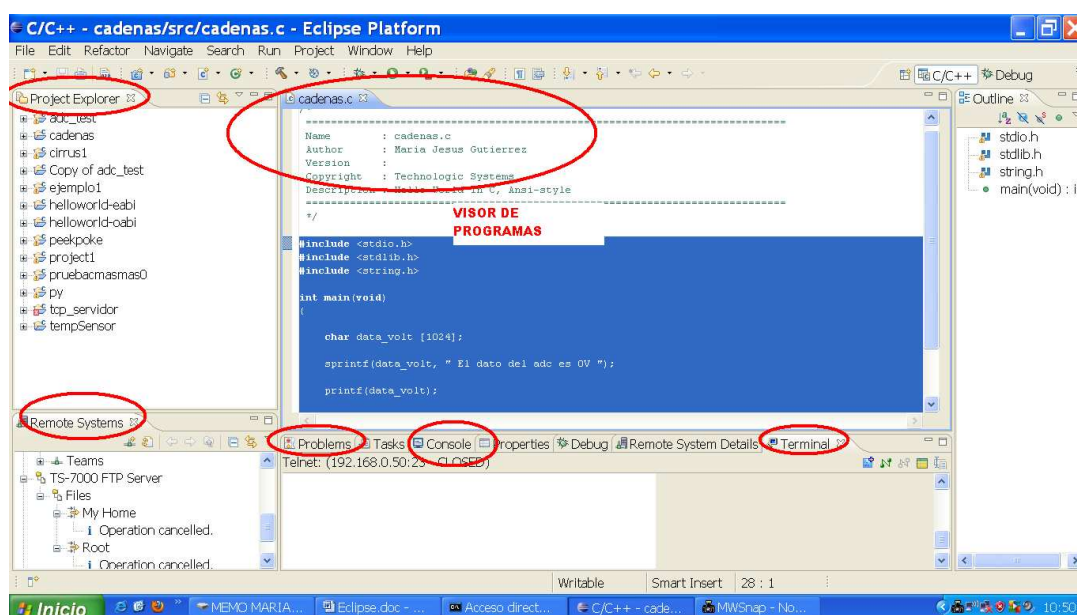


Ilustración 78: Ventanas más importantes

Las vistas más interesantes a la hora de desarrollar un proyecto para la tarjeta TS-7200, aparecen resaltadas en la Ilustración 78.

### **Project Explorer**

Proporciona una visión jerárquica de los proyecto que creados con Eclipse. Situando el cursor sobre el nombre del proyecto y haciendo click en él y sobre la carpeta *src* se puede ver los archivos. Haciendo click sobre ellos aparecerán en lo que se ha llamado *Visor de programas* en la Ilustración 78, el código del proyecto.

### Remote System

Su función es transferir el ejecutable creado en Eclipse pero no admite contraseñas en blanco para el *usuario root* como es el caso de la tarjeta TS-7200, por lo que se ha sustituido por Fillezilla que si permite esta opción.

### Console

La pestaña *Console* muestra primero los errores de configuración y una vez solucionados estos, los errores de compilación. Cuando los errores de compilación están solucionados avisa al usuario de que el proyecto ha sido compilado satisfactoriamente.

### Terminal

Establece comunicación con la tarjeta mediante el protocolo de red Telnet y realiza la misma función que Putty puede ser útil cuando se está probando el programa reiterativamente ya que dispone de los botones *Connect* y *Disconnect* para conectar y desconectar la comunicación con la tarjeta TS-7200.

### Problems

Detalla información ante los fallos que se producen durante la compilación.

## 15.1.2 Crear un workspace nuevo o cambiar de workspace

Para crear un *workspace* se hace click sobre **File** en la barra del menú principal: **File**→**Switch workspace**→**Other**. En **Browse** se crea una carpeta (**make new folder**) o se selecciona el workspace que se necesite.

## 15.1.3 Creación de un proyecto en eclipse

Para crear un nuevo proyecto hacemos click sobre **File** de la barra principal de herramientas: **File** → **New** → **C Project** de acuerdo con la Ilustración 79.

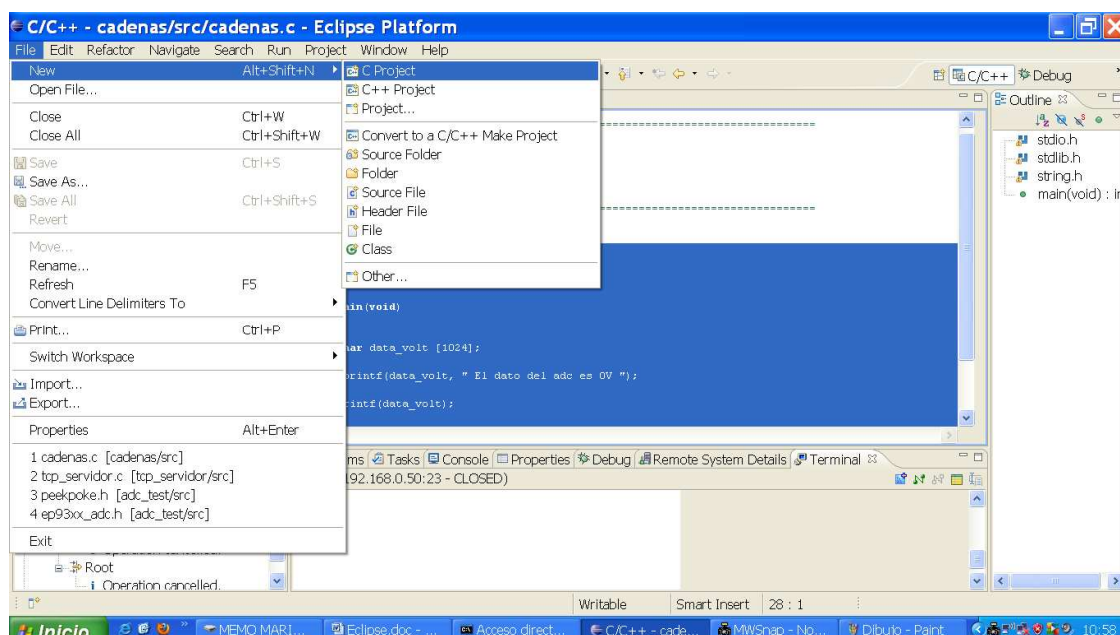
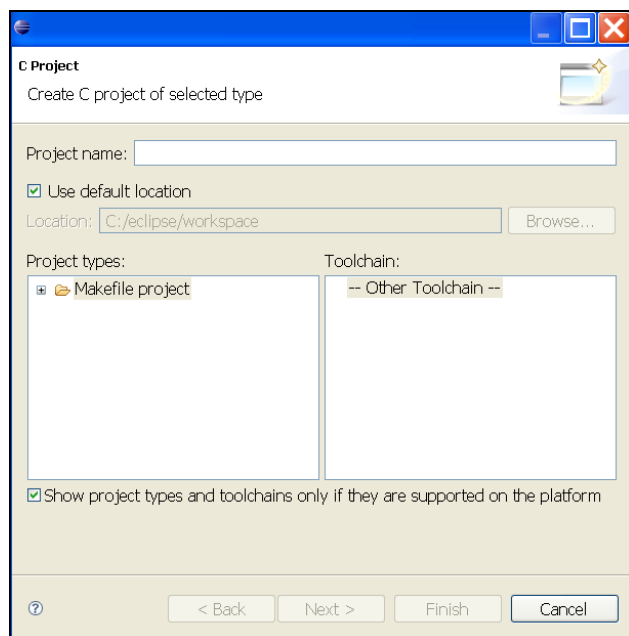


Ilustración 79: Creación nuevo proyecto

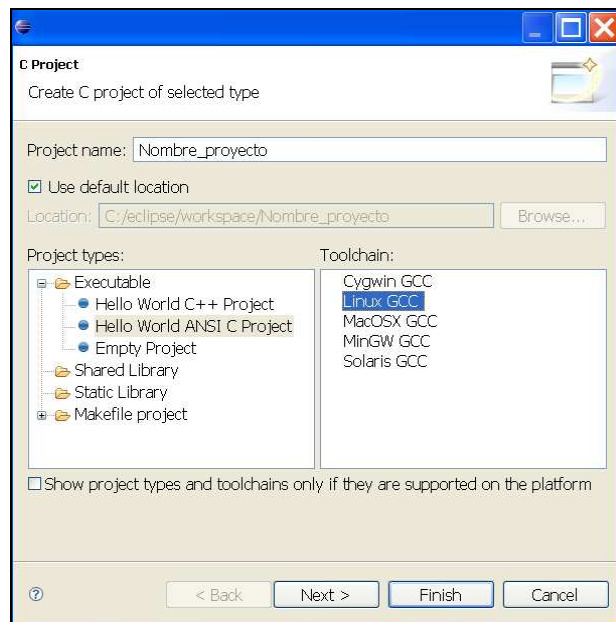


Una vez seleccionado **C Project** aparecerá la ventana de la Ilustración 80, donde se escribe el nombre del proyecto (**Project name**) y se desmarca la casilla **Show project types and toolchains if they are supported on the platform**.

En el recuadro **Project types**: Se selecciona **Hello World ANSI C Project** y en el recuadro **Toolchain**: seleccionamos **Linux GCC**, tal y como aparece en la siguiente Ilustración 81. Una vez hecho esto, presionar el botón **Next**.

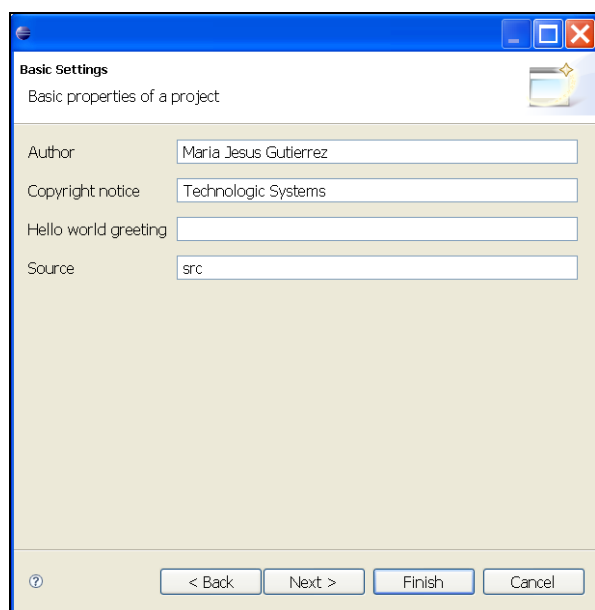


**Ilustración 80: Ventana nuevo proyecto**



**Ilustración 81: Ejemplo creación nuevo proyecto**

La carpeta *Executable*, contiene tres plantillas donde los *makefile* son generados automáticamente y proporcionan un ejecutable. Los archivos *makefile* [97] son archivos de texto que incluyen declaraciones de comentarios, variables, reglas implícitas y explícitas para más información consultar referencias: [98] [99].



**Ilustración 82: Completar con el nombre del autor es opcional**

Si se desea, en la siguiente ventana (Ilustración 82) se puede introducir el nombre del Autor y el resto de los campos, una vez introducidos y se presiona de nuevo el botón **Next**. Los datos introducidos aparecerán reflejados automáticamente en el proyecto una vez se haya creado. Completar esta pestaña es opcional y si se completa se verá reflejada automáticamente en el código.

### 15.1.3.1 Selección de las configuraciones de un proyecto

En la siguiente ventana de la Ilustración 83, se seleccionan las configuraciones *Debug* y/o *Release* (por defecto están seleccionadas ambas configuraciones) y se presiona el botón *Advanced settings*. Con *Debug* y *Release* seleccionado se pulsa el botón *Advanced settings* para realizar la configuración adecuada a la tarjeta TS-7200 y llegar a la ventana llamada: *Properties for* seguido del nombre del proyecto (Ilustración 84).

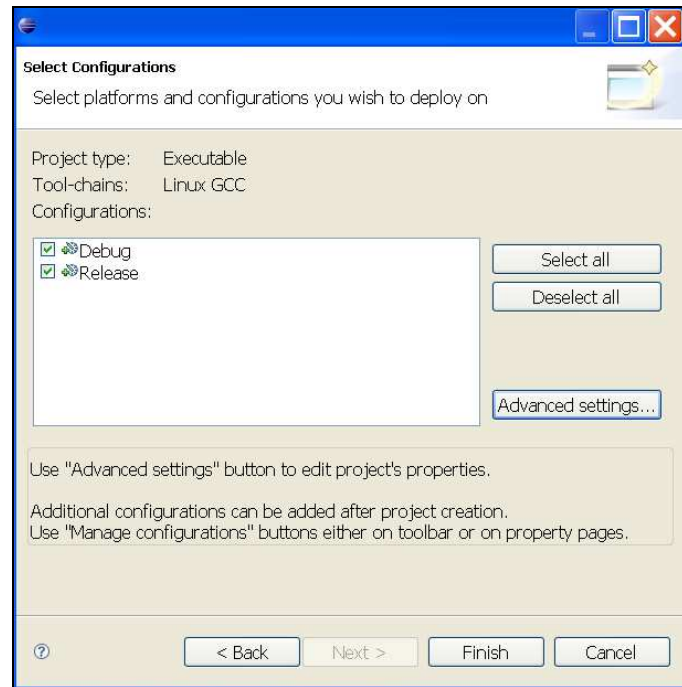


Ilustración 83: Selección del botón *Advanced settings*

### 15.1.3.2 Configuración de las propiedades un proyecto

Ya en la ventana *Properties for* nombre del proyecto (Ilustración 84), se selecciona *C/C++ Build* en la columna situada a la izquierda. En la pestaña *Builder settings*, se selecciona del desplegable de *Builder type* seleccionamos *Internal builder*.

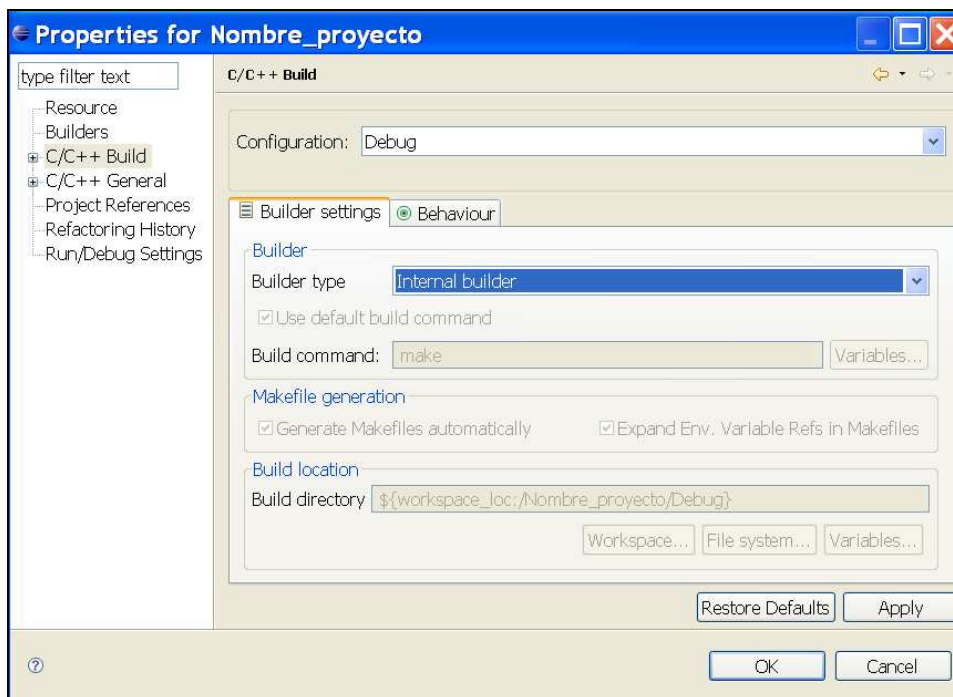
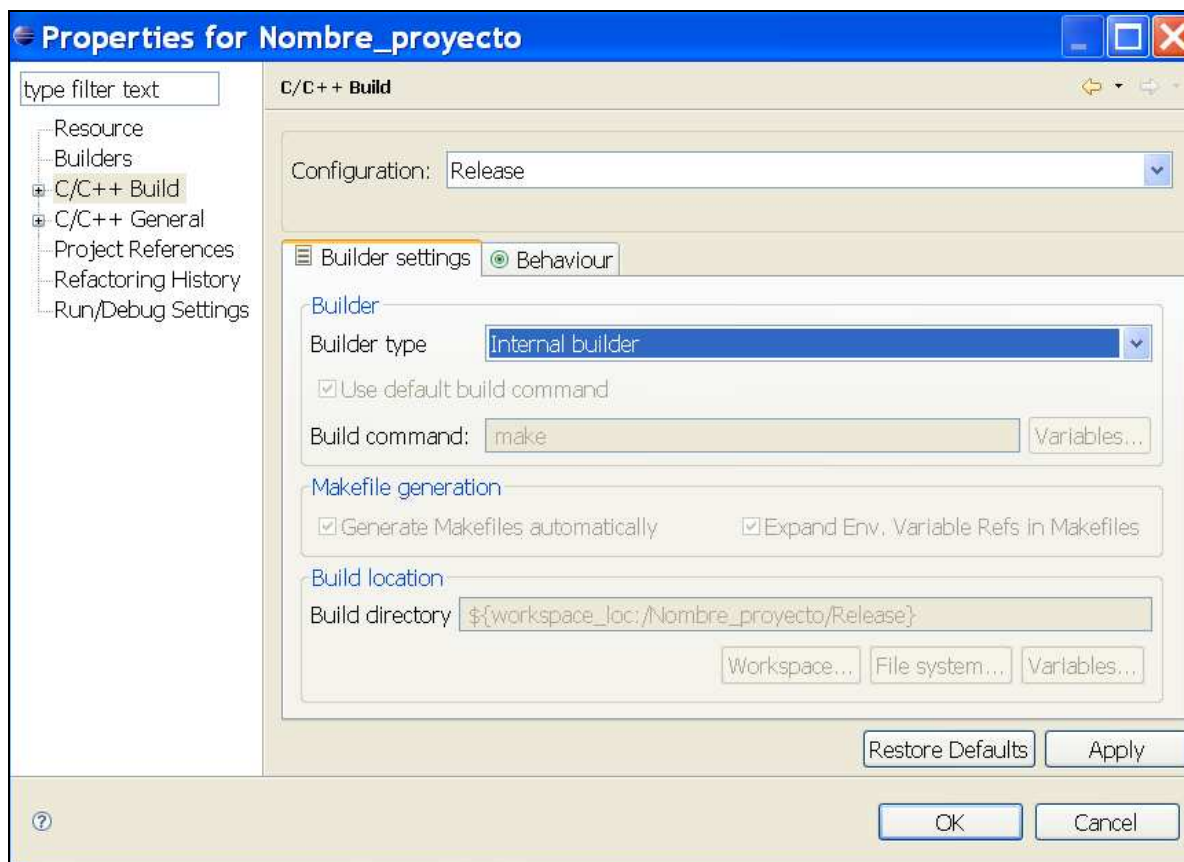


Ilustración 84: Selección *Internal builder/Debug*

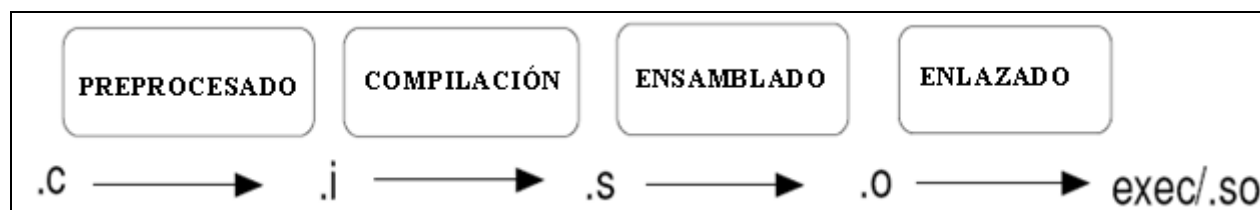
Después se cambia el desplegable de **Configuration** de **Debug** a **Release** y en el desplegable de **Builder type** de la pestaña **Builder settings** se selecciona, de nuevo, **Internal builder** tal y como aparece en la Ilustración 85.



**Ilustración 85: Selección Internal builder/Release**

Para llegar a la ventana de la Ilustración 87, se selecciona en la columna de la izquierda **Discovery options** (marcado con una flecha roja en la Ilustración 87) dentro de **C/C++ Build**. En el desplegable de **Configuration** se selecciona **Debug** y en **Compiler invocation command** sustituimos **gcc** por: **arm-unknown-linux-gnu-gcc.exe**

**GCC (GNU Compiler Collection)** [100] es un compilador integrado del proyecto GNU para C y otros lenguajes como C++ o Fortran. Permite a partir de un programa fuente escrito en lenguaje C en este caso, generar un programa ejecutable binario en el lenguaje de la máquina donde ha de ejecutarse. Los pasos que realiza GNU para obtener el ejecutable aparecen en la Ilustración 86 [101], para ampliar información sobre este tema, consúltese las referencias [101] y [102].



**Ilustración 86: Pasos para obtener un ejecutable desde el código fuente .C**

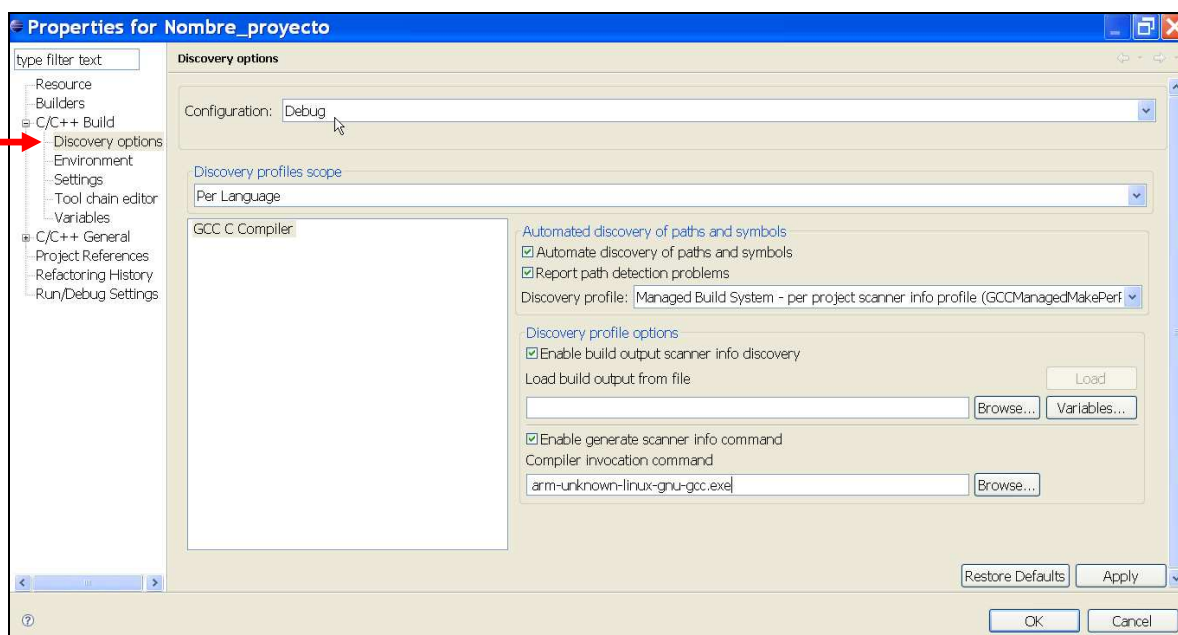


Ilustración 87: Compiler invocation command Debug

Análogamente se modifica en el desplegable de *Configuration Release* y en *Compiler invocation command* sustitimos *gcc* por: **arm-unknown-linux-gnu-gcc.exe** (ver Ilustración 88)

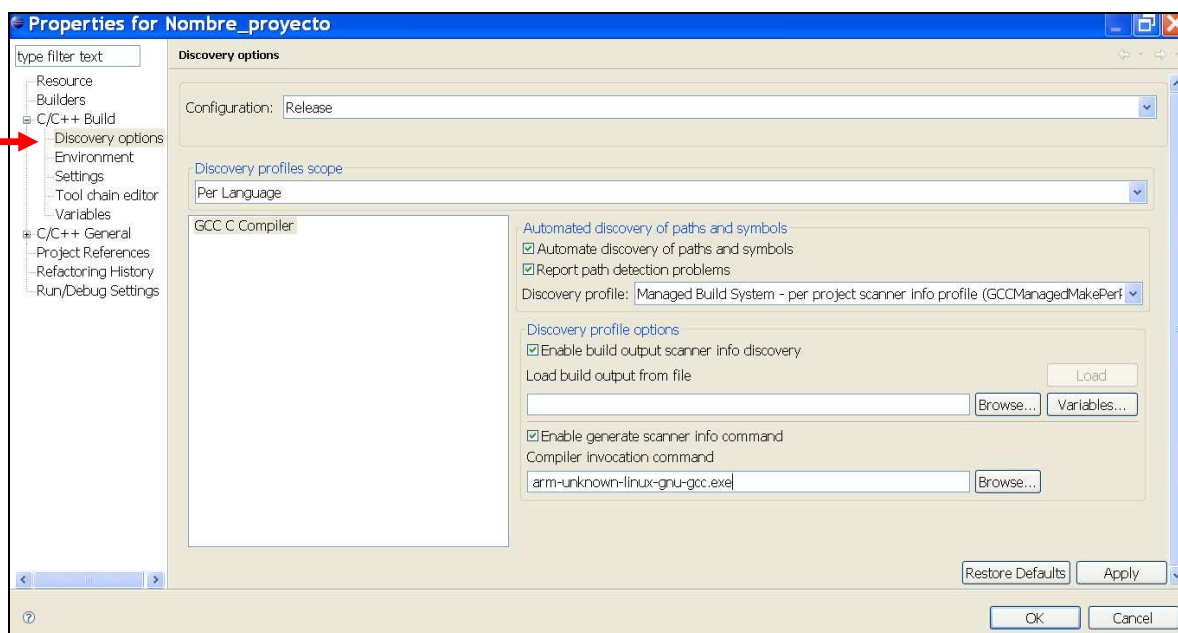


Ilustración 88: Compiler invocation command Release

Se selecciona ahora de la columna de la izquierda llamada **Settings** dentro de *C/C++ Build* (ver Ilustración 89), se realizarán los cambios siguientes para la configuración Debug (en *Configuration* debe seleccionarse en el desplegable **Debug**). En la pestaña *Tool settings* se hace click en **GCC C Compiler** y en el recuadro *Command* se sustituye *gcc* por **arm-unknown-linux-gnu-gcc.exe**



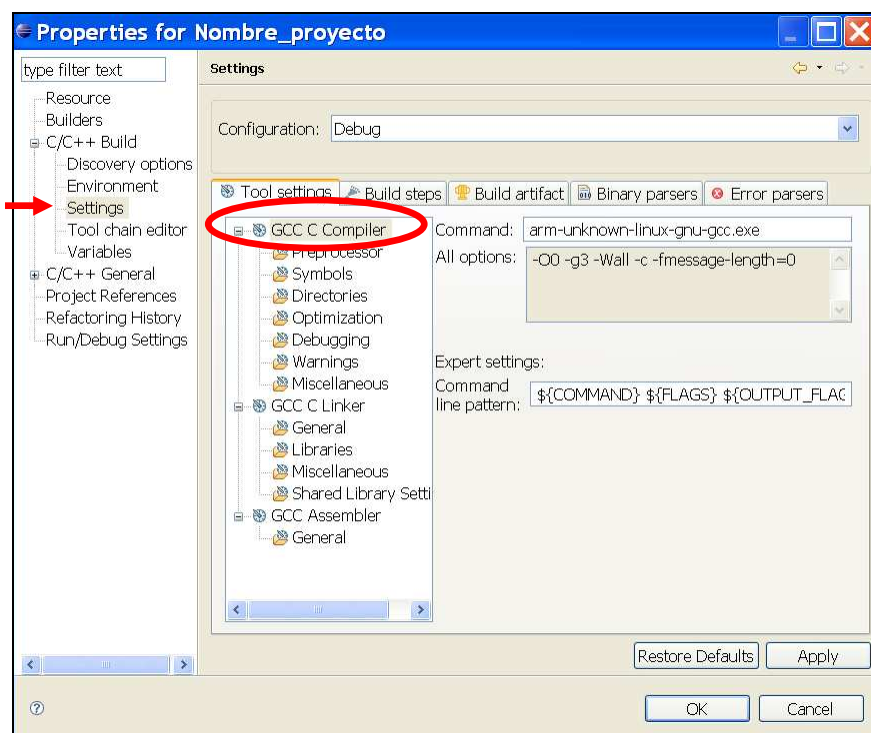


Ilustración 89: GCC C Compiler Debug

A continuación se hace click en ***GCC C Linker*** y en el recuadro Command se sustituye *gcc* por **arm-unknown-linux-gnu-gcc.exe** conforme a la Ilustración 90.

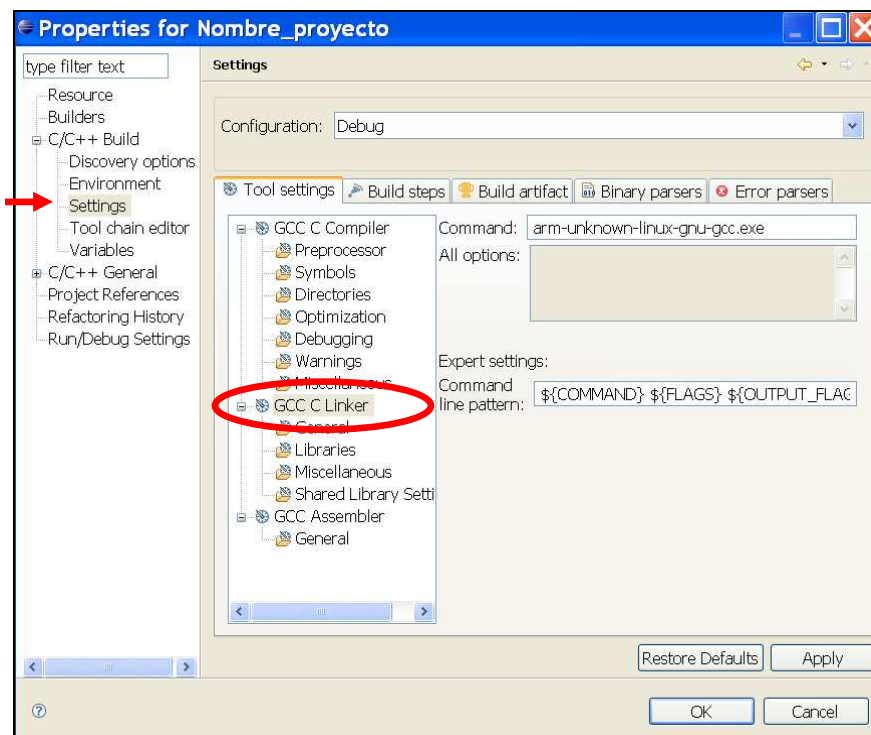


Ilustración 90: GCC C Linker Debug

Por último se hace click en ***GCC Assembler*** (ver Ilustración 91) y en el recuadro Command se sustituye *as* por **arm-unknown-linux-gnu-as.exe**

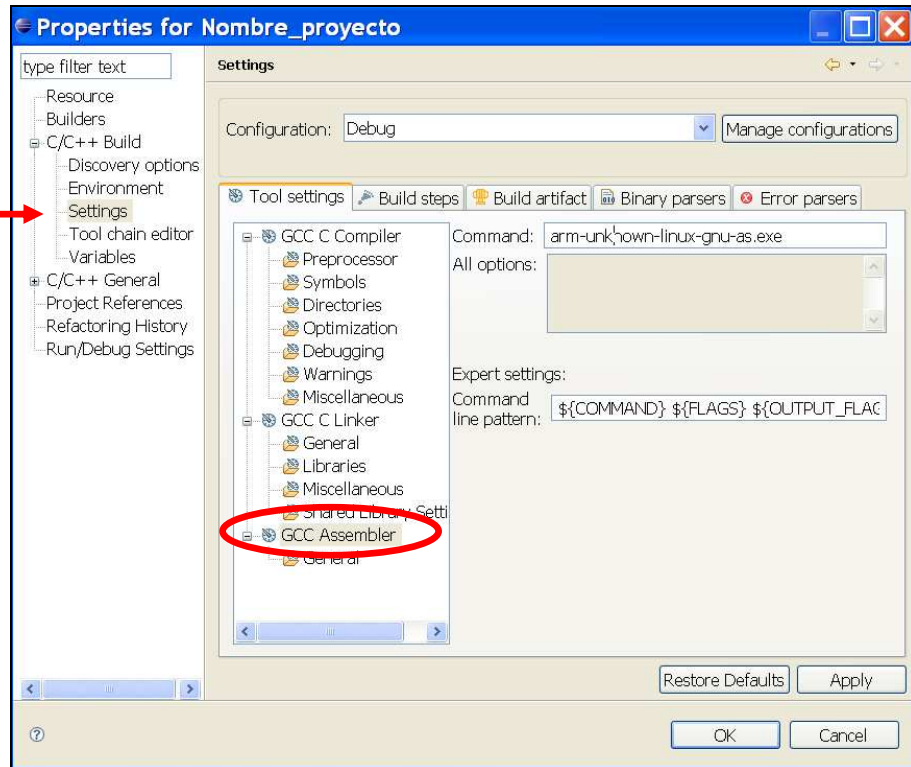


Ilustración 91: GCC Assembler Debug

Análogamente, se repite el mismo proceso para la configuración Release (en el recuadro *Configuration*, se selecciona **Release** en el desplegable y se repiten los pasos: *GCC C Compiler*, *GCC Linker* y *GCC Assembler*, de acuerdo con las ilustraciones 92, 93 y 94).

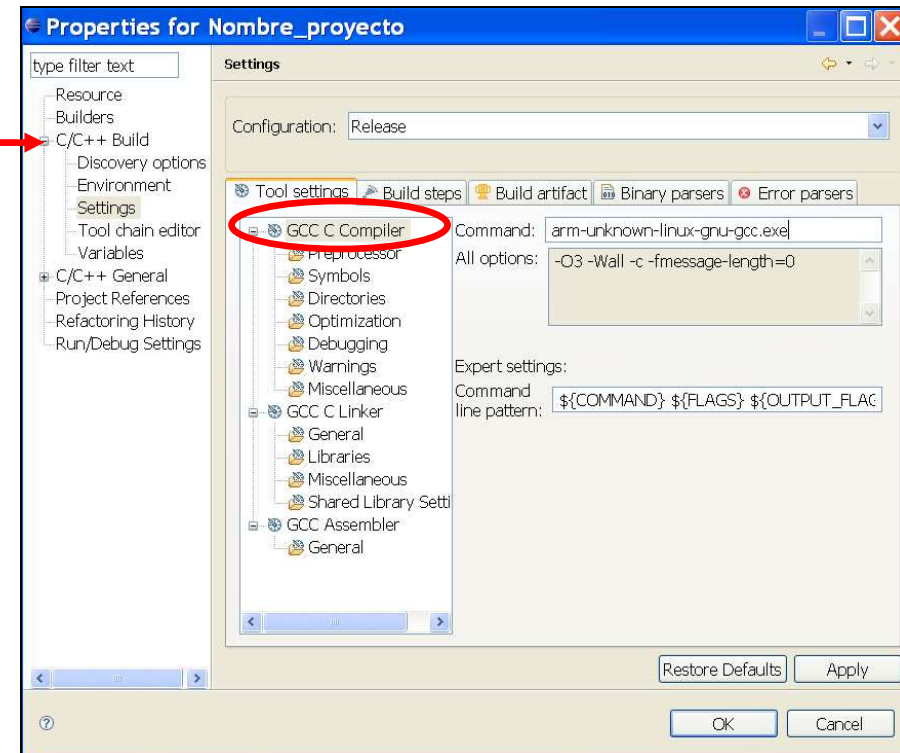


Ilustración 92: GCC C Compiler Release

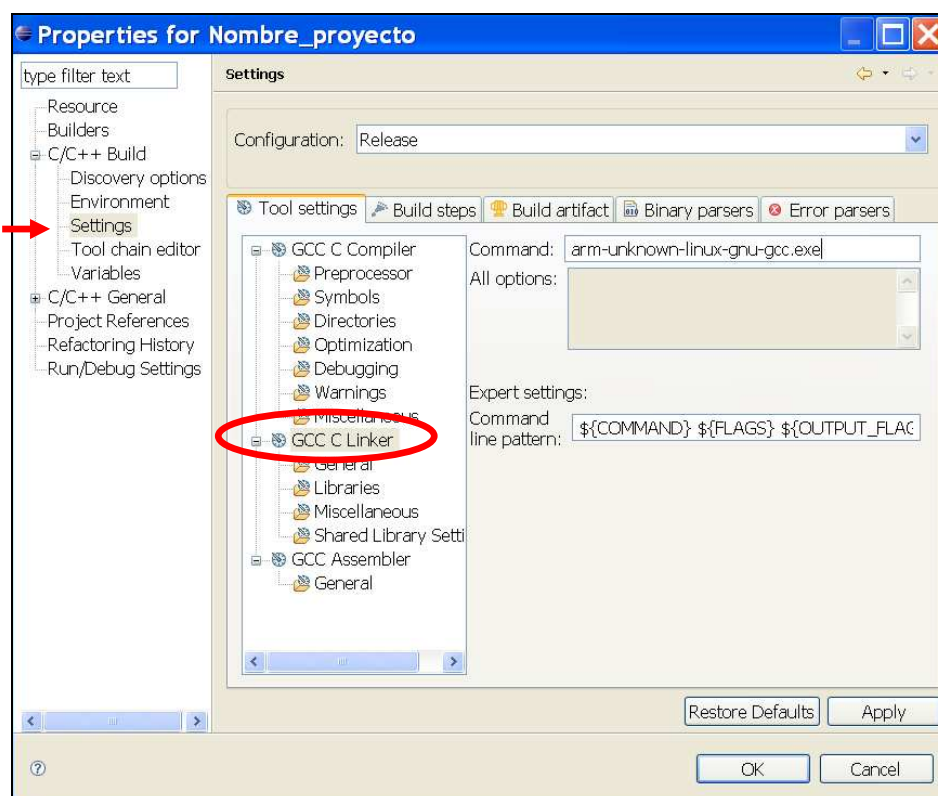


Ilustración 93: GCC C Linker Release

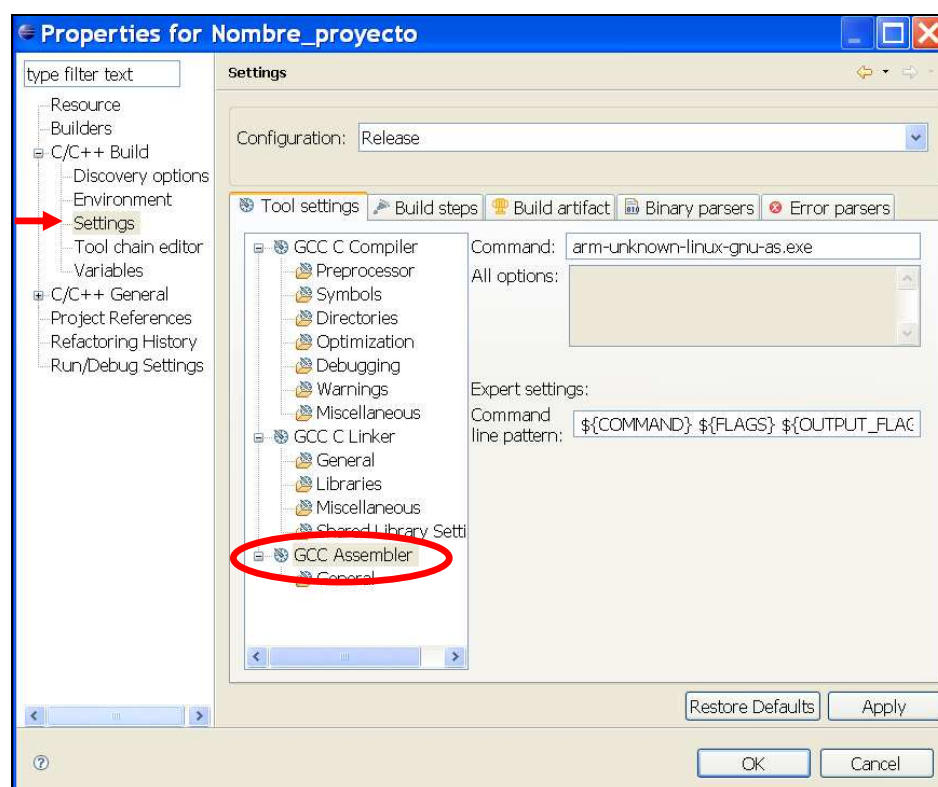


Ilustración 94: GCC Assembler Release

Nota: Si los cambios realizados, no se hicieran efectivos: introducir manualmente las instrucciones, *arm-unknown-linux-gnu-gcc.exe* y *arm-unknown-linux-gnu-as.exe*, o al menos una letra manualmente en lugar de introducirlos mediante las herramientas copiar y pegar.

Una vez realizado los cambios, pulsar el botón **OK**. No se aconseja utilizar el botón **Apply**, ya que puede cambiar la configuración que se acaba de introducir. Al seleccionar **OK** aparece de nuevo la ventana *Select Configuration* (ver Ilustración 95) donde se selecciona **Finish** para crear el nuevo proyecto.

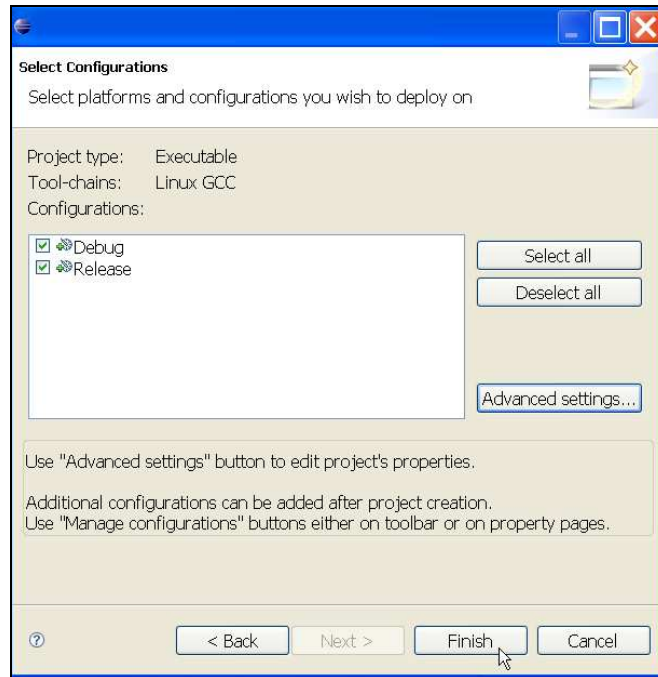


Ilustración 95: Ventana Select Configuration

## 15.1.4 Compilación

Una vez realizada la creación de un proyecto nuevo configurado de acuerdo con el apartado 15.1.3 aparecerá en la vista *Project Explorer* una carpeta con el nombre del proyecto creado, como se aprecia en la Ilustración 96.

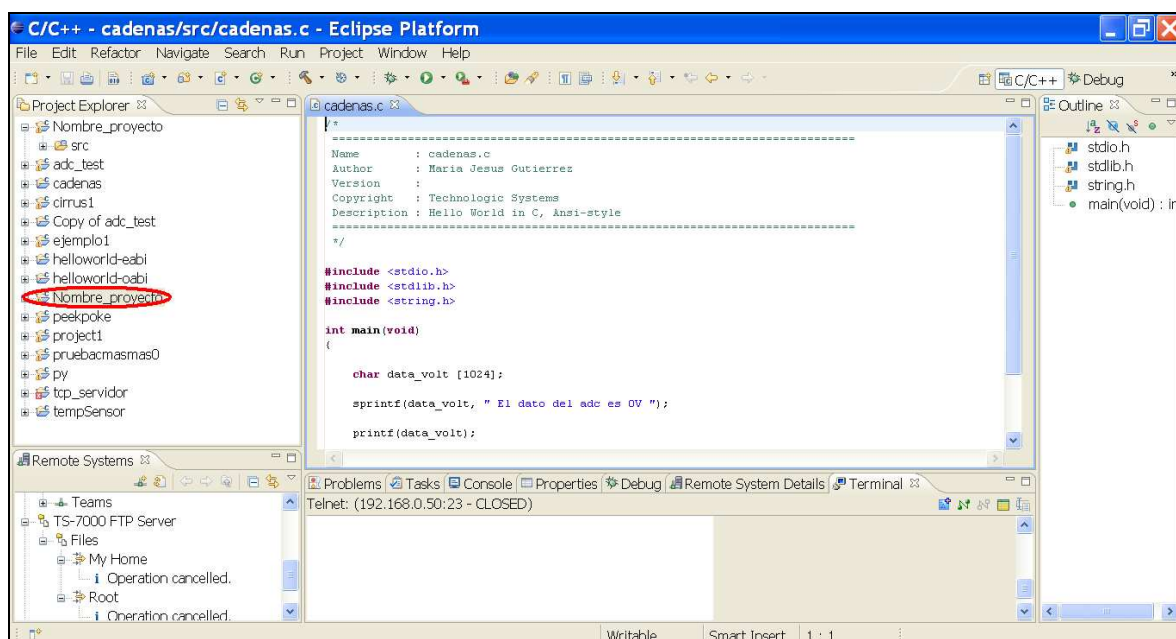


Ilustración 96: Nuevo proyecto creado

La carpeta con el nombre del archivo creado contiene una carpeta de nombre *src* (ver Ilustración 97) donde aparece un archivo *.c*. Es en este archivo, donde se sustituye el código que aparece por defecto (ver Ilustración 98) durante la configuración por el código deseado.

En la Ilustración 99 se ha escrito un código ejemplo para explicar la compilación.

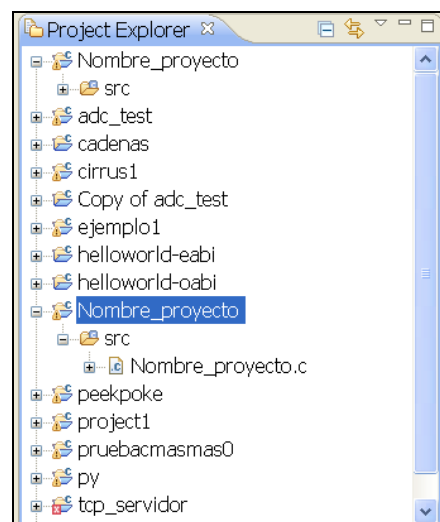


Ilustración 97: Detalle del nuevo proyecto

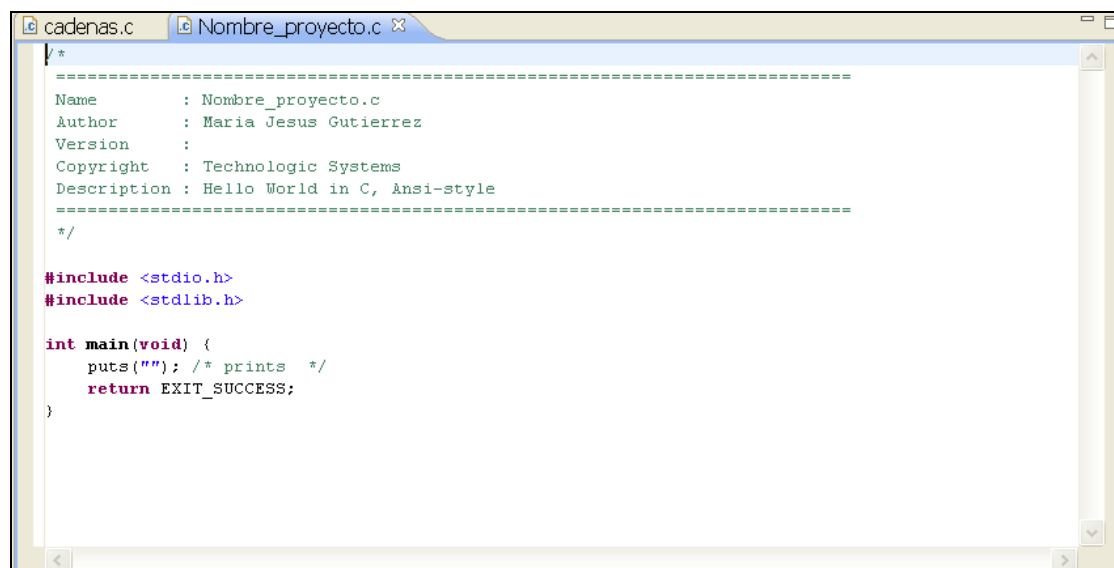


Ilustración 98: Código que aparece por defecto

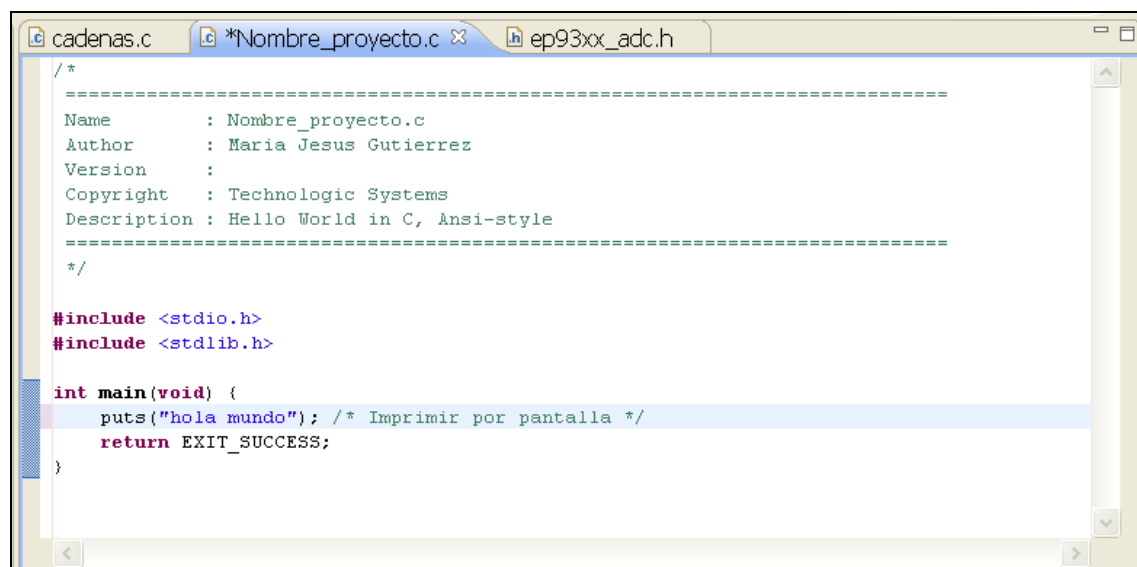


Ilustración 99: Código Ejemplo



Una vez el código esté terminado, se realizará la compilación, pero antes se debe **guardar siempre el proyecto**, de no ser así, Eclipse compilará el proyecto anterior a las modificaciones, y no se recibirá ningún aviso al respecto.

El proyecto se puede guardar, pulsando el botón *Save* de la barra de herramientas que aparece remarcado en la Ilustración 100 o mediante el comando Ctrl+S.

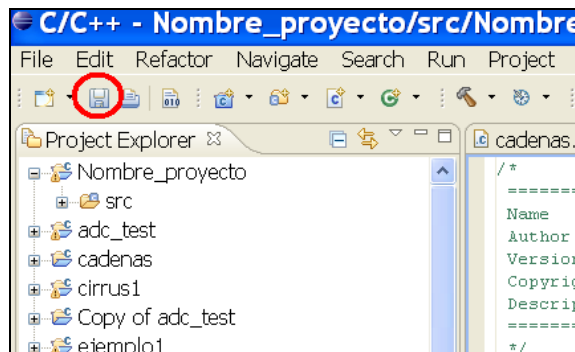


Ilustración 100: Guardar el proyecto

Una vez guardado se sitúa el cursor sobre la carpeta con el nombre del proyecto en la vista **Project Explorer** y haciendo click en el botón derecho del ratón aparece el desplegable de la Ilustración 101. Se selecciona **Build Configuration** en el menú del desplegable y después:

**Build Configuration → Build → all**

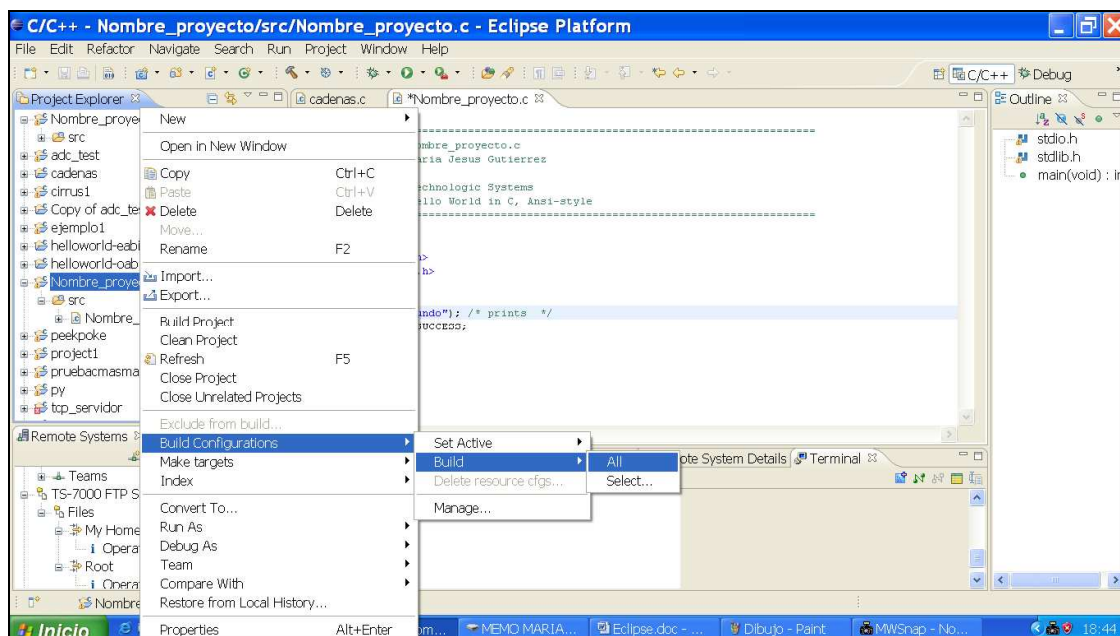


Ilustración 101: Compilar

El resultado de la compilación se muestra en la pestaña **Console**. Si todo es correcto el mensaje que debería aparecer es similar al que aparece en la Ilustración 102:

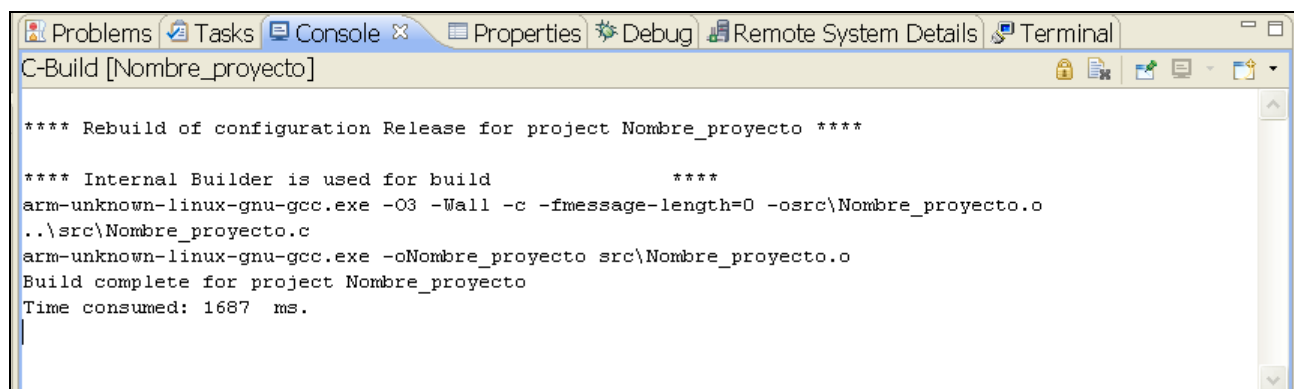


Ilustración 102: Mensaje de compilación de código sin errores

En el caso de que se produjeran errores durante la compilación, Eclipse presenta primero, los fallos de compilación debidos a la configuración y después a los errores propios de la compilación del programa. Los primeros son aquellos que se deben a una configuración incorrecta, bien porque el compilador elegido no es válido o bien por un fallo a la hora de escribir el compilador, esto último es bastante habitual cuando se produce un error de este tipo. Para corregirlos se sitúa el ratón sobre carpeta con el nombre del proyecto en la vista **Project Explorer**, y con el botón derecho del ratón se selecciona del desplegable la opción **Properties** que devuelve al usuario a la ventana **Properties for** nombre del proyecto. Para solucionar el problema será necesario repetir los pasos para la configuración de las propiedades del proyecto (apartado 15.1.3.2) y volver a compilar.

Después aparecerán los fallos de compilación del programa. Para resolverlos puede ser de ayuda la pestaña **Problems** donde aparece una descripción de los errores y su localización. Una vez corregidos los errores se vuelve a compilar el proyecto. Cuando después de compilar, en la pestaña **Console** aparece el mensaje: “**Build complete for Project**” indica que el proceso de compilación ha finalizado con éxito. En ese momento en la carpeta con el nombre del proyecto que figura en la vista **Project Explorer** aparecerán dos carpetas nuevas junto a la carpeta **src**: **Debug** y **Release** como se puede observar en la Ilustración 103.

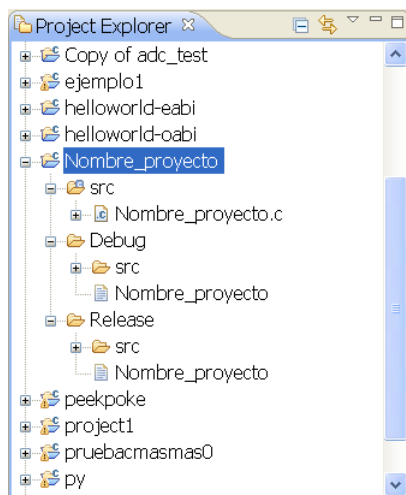


Ilustración 103: Detalle de nuevo proyecto compilado

### 15.1.5 Incluir librerías en el proyecto

Si se han incluido llamadas a librerías en el código y no se añaden las librerías al proyecto el compilador dará error. Para añadir las librerías se selecciona la carpeta **src** dentro del proyecto y se hace click en: **File->New->Header File**.

En la ventana **New Header File** se completa el recuadro **Header File** con el nombre de la librería y al pulsar el botón **Finish** aparecerá el archivo en la carpeta seleccionada previamente. Se hace doble click sobre el fichero y el documento se abre en el visor de programa, es en este archivo donde se escribe la librería. Una vez incluidas las librerías se procede a compilar.

## 15.2 FILEZILLA

Para trasladar el ejecutable creado por Eclipse a la tarjeta se utiliza el programa FileZilla cuyo logotipo aparece en la Ilustración 104. FileZilla es un gestor de FTP gratuito que se puede obtener descargándolo de la página oficial [103]. En la referencia [104] se puede obtener la documentación necesaria su instalación.



Ilustración 104: Logotipo FileZilla

### 15.2.1 Acceso al programa

El acceso al programa se realiza en: **Programas** → **FileZilla FTP Client** → **FileZilla**

### 15.2.2 Entorno

La ventaja del uso de FileZilla frente a la pestaña *Remote Systems* de Eclipse es que a la hora de conectarse con la tarjeta permite introducir como usuario *root* y ninguna contraseña. Mientras que Remote System exige introducir una contraseña siempre y la tarjeta TS-7200 no dispone de contraseña inicialmente, para el usuario *root*. Se ha preferido mantener la tarjeta sin contraseña para facilitar el acceso en futuras aplicaciones. Otra ventaja es que al trabajar con dos programas distintos, Eclipse responde mejor.

En el entorno de FileZilla se distinguen tres sectores: la *localización*, el *remoto* y la *comunicación* que se encuentran remarcados en la Ilustración 105.

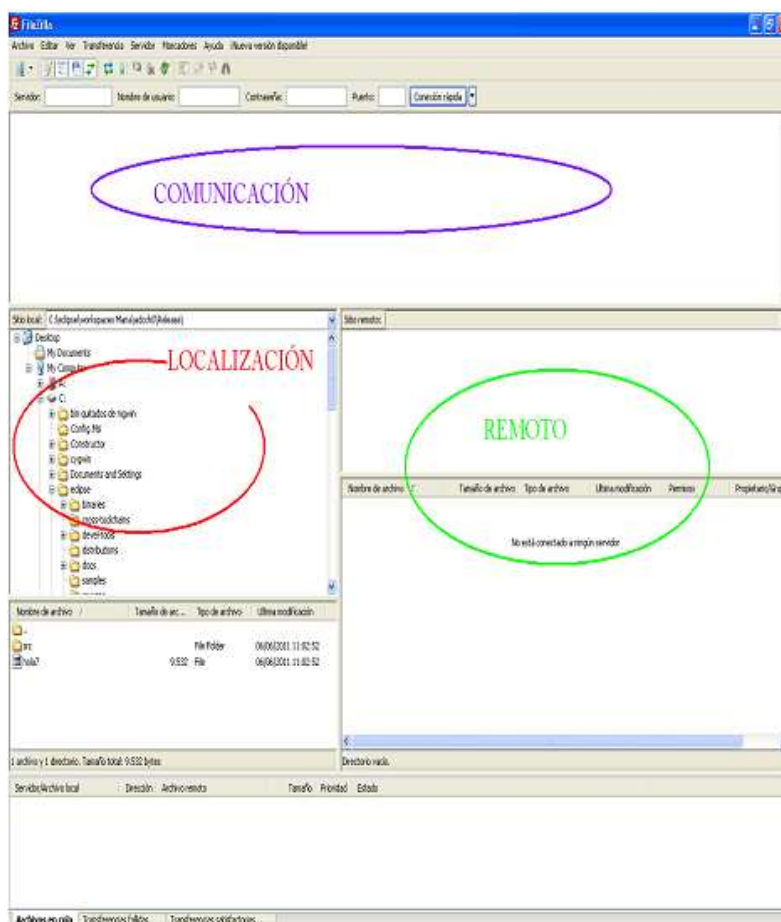


Ilustración 105: Entorno FileZilla

La *localización* corresponde al árbol del PC en el que está instalado FileZilla. En el *remoto* se encuentra el árbol de la tarjeta TS-7200. Y en *comunicación* se puede encontrar mensajes sobre el estado de conexión con la tarjeta TS-7200.



### 15.2.2.1 Localización

Dentro de Localización se encuentra el recuadro **sitio local** (señalado con una flecha en la Ilustración 106) donde se buscarán los proyectos en el workspace que se hayan creado en Eclipse (si no ha creado ningún workspace nuevo, aparecerán, el carpeta dentro de Eclipse llamada *workspace*). Si se sitúa el cursor sobre el proyecto en la ventana inferior, en **nombre de archivo** aparecerá el contenido de las carpetas.

Nota: Se debe recordar a la hora de buscar los proyectos que el programa Eclipse debe estar guardado en el directorio C, según las instrucciones de instalación del mismo.

En el ejemplo de la Ilustración 106, el proyecto `adcch0` está guardado en el workspace *workspaces María* previamente creado siguiendo las indicaciones del apartado 15.1.2. Al compilar el proyecto correctamente, aparecerán tres carpetas:

- **Debug**
- **Release**
- **src**

En realidad la aparición de las carpetas *Debug* y *Release* depende de cómo se haya realizado la configuración (consultar apartado 15.1.3.1), por lo que podría no aparecer alguna de ellas.

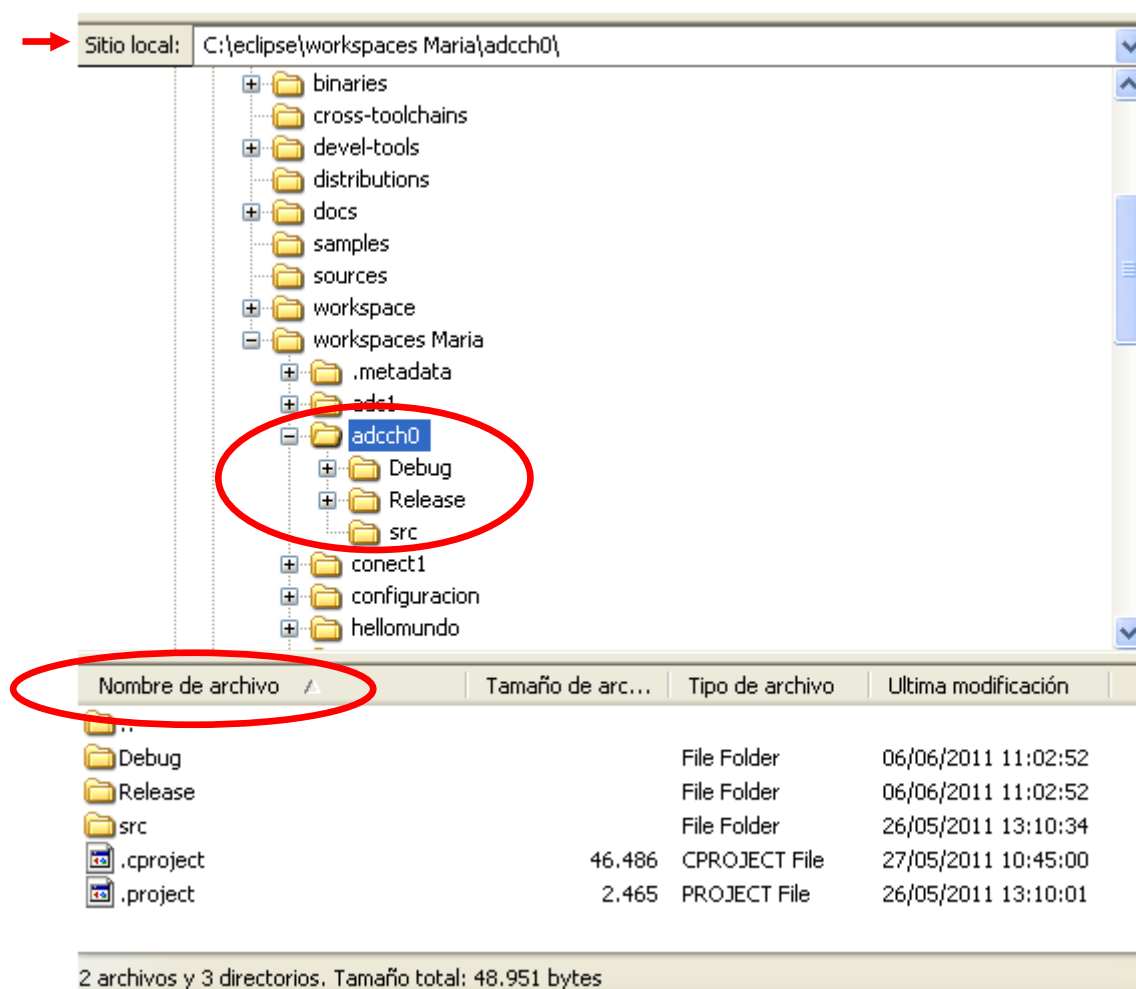


Ilustración 106: Localización

La clave está, en que en la carpeta **Release**, será ahí donde aparecerá el ejecutable del proyecto que trasladaremos a la tarjeta TS-7200. En el ejemplo anterior: En la carpeta Release encontraremos el proyecto hola7 (ver Ilustración 107). Es importante fijarse en el tamaño de archivo si es excesivamente grande es posible que se haya producido un error, en ese caso repetir el proceso de compilación y si persiste revisar fallos en la configuración o en la compilación.

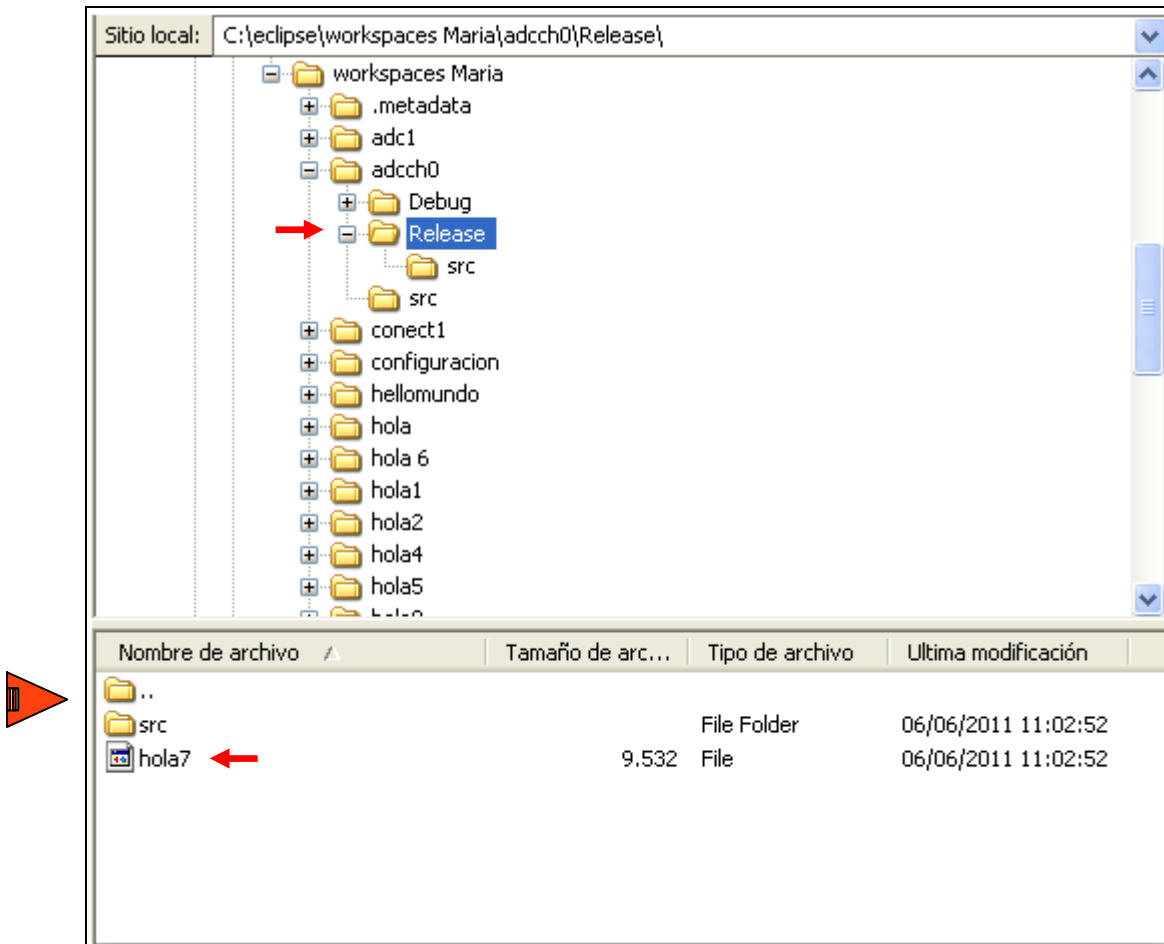


Ilustración 107: Localización - carpeta Release

## 15.3 CONEXIÓN A LA TARJETA TS-7200

Para conectar con la placa se ha realizado el siguiente procedimiento:

1. Se procederá en primer lugar a alimentar la placa con un voltaje de 5V. Es importante que la alimentación no superare los 6V y que tampoco sea inferior a los 4V. Si la tensión cae por debajo del mínimo de tensión admisible en la tarjeta se encenderían el led verde y el rojo, que hay junto a la entrada de la alimentación, simultáneamente. Solución: Aumentar la tensión de alimentación entre los límites aceptables para que la tarjeta pueda funcionar.
2. Conectar la placa al PC mediante el cable de Ethernet.
3. Establecer comunicación con la tarjeta mediante protocolo FTP en el FileZilla para ello, ir al menú principal del programa FileZilla, tal como muestra la Ilustración 108, y en

**Archivo:** *Archivo* → *Gestor de sitios* o utilizar el comando Ctrl+S para acceder a la ventana *Gestor de sitios* (ver Ilustración 108) donde se selecciona el botón **nuevo sitio**.

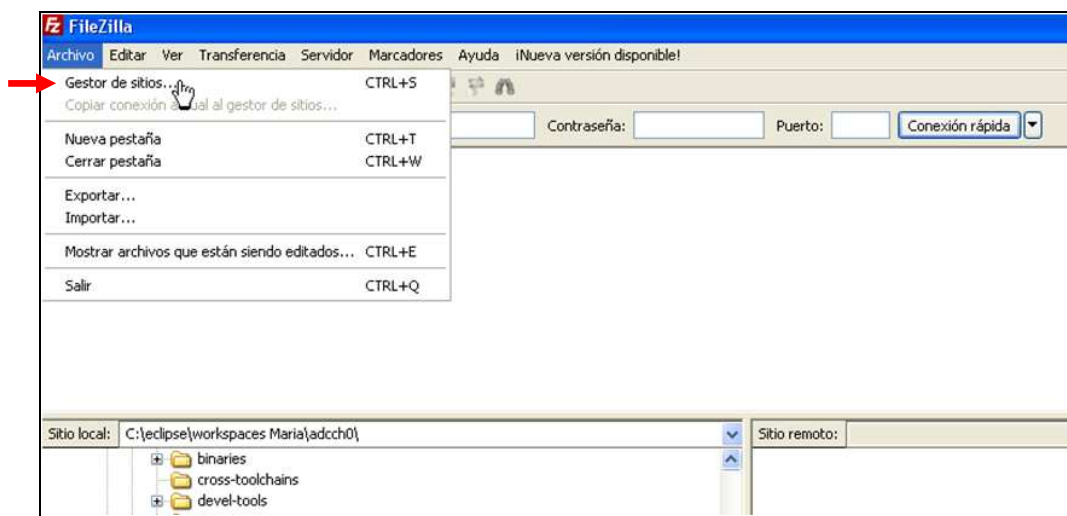


Ilustración 108: Gestor de sitios

Al seleccionar **nuevo sitio** (ver Ilustración 109) automáticamente, los campos *Servidor* y *Puerto* permiten el acceso a escritura de los mismos. A continuación se rellena el campo llamado *Servidor* (ver Ilustración 110) con la dirección IP de la tarjeta que por defecto es 192.168.0.50 [105].

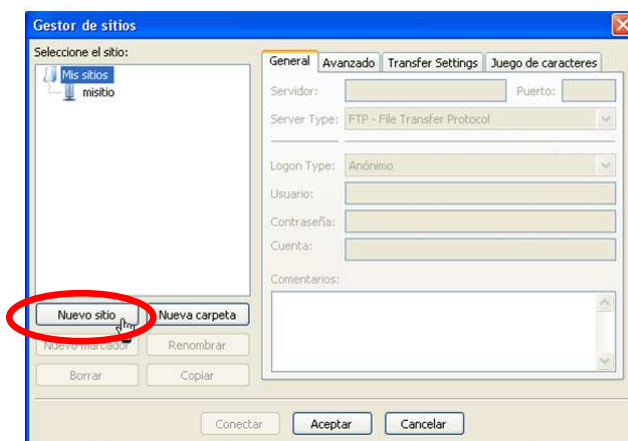


Ilustración 109: Ventana Gestor de sitios

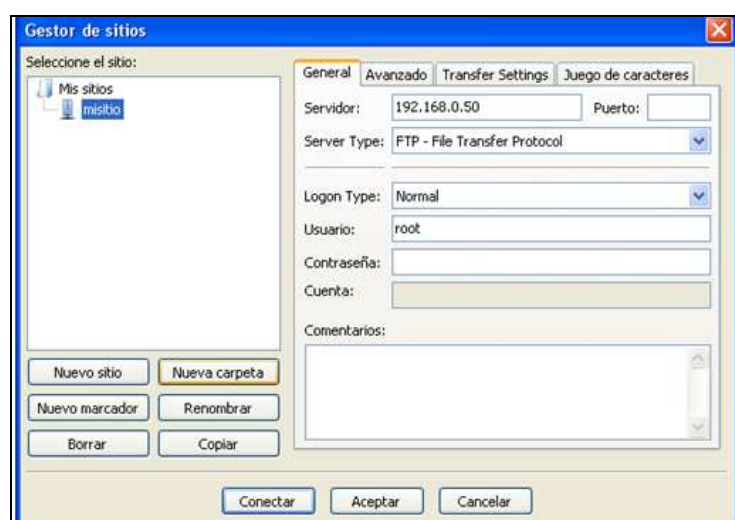


Ilustración 110: Como completar la ventana Gestor de sitios

Al seleccionar la opción **Normal** en el desplegable *Logon Type* (Ilustración 110) se hacen accesibles los campos de usuario y contraseña que se completan de forma que:

- En *Usuario* → la palabra *root*
- Y en *Contraseña* → no se rellena, se dejará en blanco.

Por último podemos personalizar el nombre al sitio que hemos creado utilizando el botón *Renombrar*. En el ejemplo de la Ilustración 110 se ha creado un sitio llamado: *misitio*. Si presionamos el botón aceptar se habrá creado un nuevo sitio.

Nota: Si no se incluye un puerto específico, FileZilla toma por defecto el puerto 21.

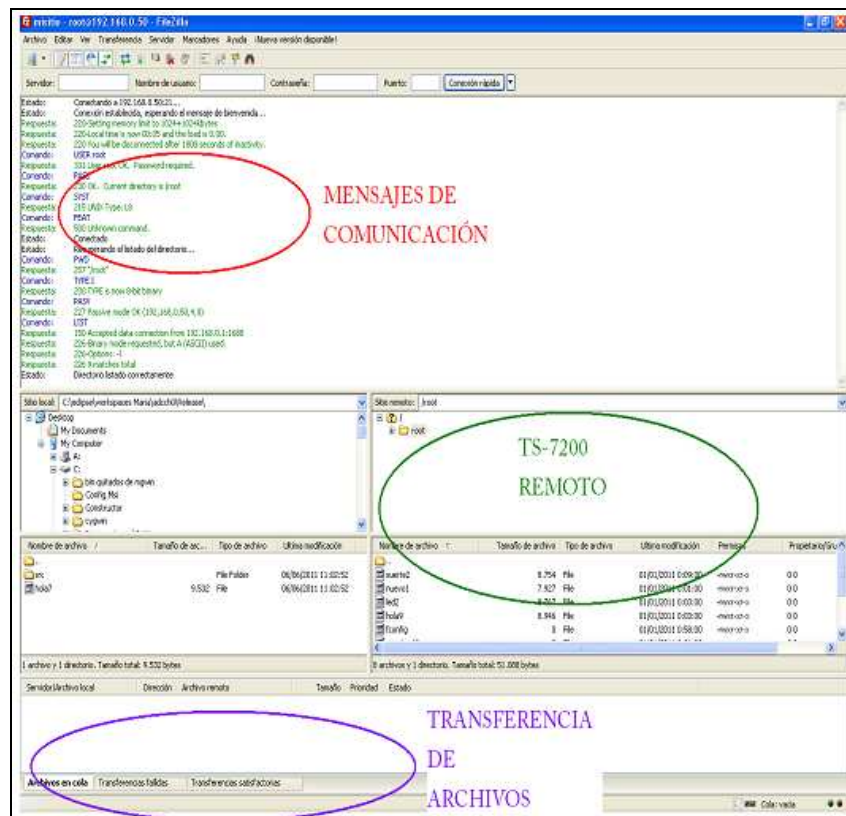
Una vez creado el sitio, si deseamos conectarnos a la placa bastará con ir a menú principal: **Archivo**→**Gestor de sitios**

En la ventana: **Gestor de Sitios** seleccionamos el botón **Conectar** y tras esperar unos segundos comienza la comunicación, a la espera del mensaje de bienvenida. Los mensajes serán similares a los que aparecen en la Ilustración 111.

```
Estado: Conectando a 192.168.0.50:21...
Estado: Conexión establecida, esperando el mensaje de bienvenida...
Respuesta: 220-Setting memory limit to 1024+1024kbytes
Respuesta: 220-Local time is now 00:05 and the load is 0.00.
Respuesta: 220 You will be disconnected after 1800 seconds of inactivity.
Comando: USER root
Respuesta: 331 User root OK. Password required.
Comando: PASS
Respuesta: 230 OK. Current directory is /root
Comando: SYST
Respuesta: 215 UNIX Type: L8
Comando: FEAT
Respuesta: 500 Unknown command.
Estado: Conectado
Estado: Recuperando el listado del directorio...
Comando: PWD
Respuesta: 257 "/root"
Comando: TYPE I
Respuesta: 200 TYPE is now 8-bit binary
Comando: PASV
Respuesta: 227 Passive mode OK (192,168,0,50,4,0)
Comando: LIST
Respuesta: 150 Accepted data connection from 192.168.0.1:1688
Respuesta: 226-Binary mode requested, but A (ASCII) used.
Respuesta: 226-Options: -l
Respuesta: 226 9 matches total
Estado: Directorio listado correctamente
```

**Ilustración 111: Mensajes de Filezilla**

- Si la comunicación se ha establecido la tarjeta TS-7200 aparecerá en el remoto. En la Ilustración 112 muestra como ha aparecido en el remoto la carpeta *root* de la tarjeta TS-7200.



**Ilustración 112: Transferencia de archivos**

## 15.4 TRANSFERENCIA DE ARCHIVOS

Una vez establecida la comunicación, la transferencia de archivos se realiza haciendo doble click sobre el archivo a transferir arrastrando el archivo deseado desde el sitio local, en este caso hola7 (ver Ilustración 113) a la carpeta destino en *root* de la tarjeta TS-7200.

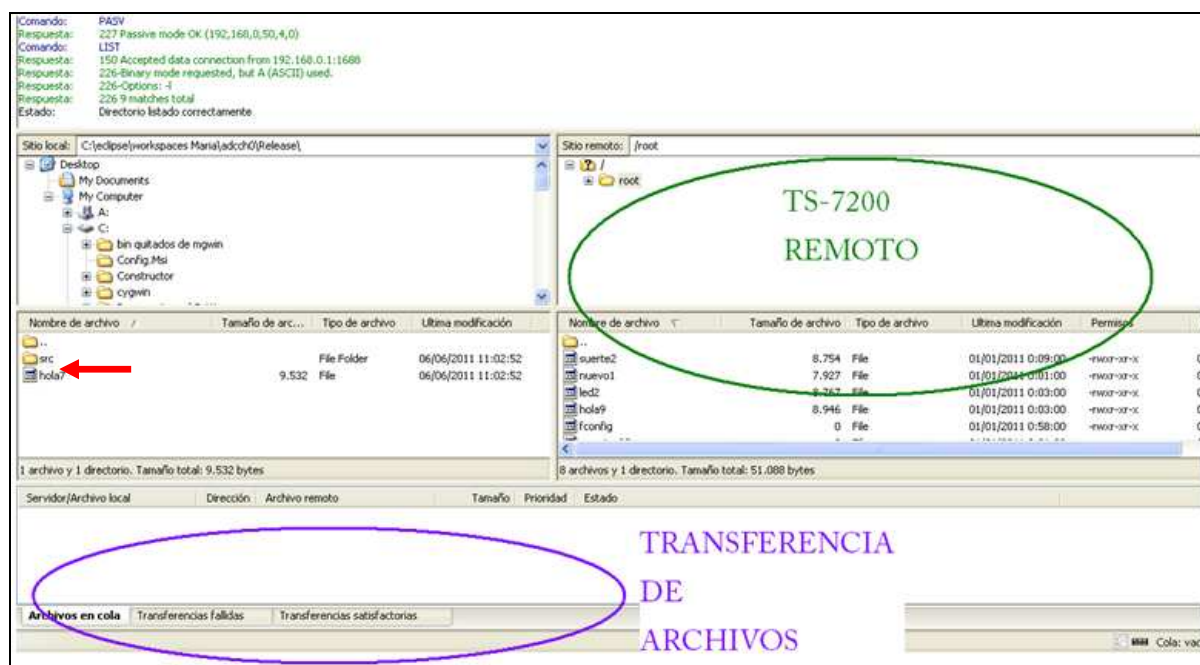


Ilustración 113: Módulos de Filezilla

La transferencia de archivos puede llevar unos segundos. En la Ilustración 114 puede verse como ahora el archivo “hola7” está ahora en la carpeta “carpeta” en *root*.

En las pestañas de *Transferencias fallidas* y *Transferencias satisfactorias* que aparecen en la Ilustración 115) se puede obtener información acerca de las transferencias.

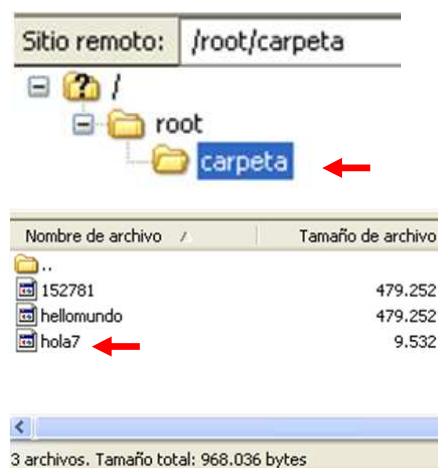


Ilustración 114: Detalle 1 transferencia de archivos

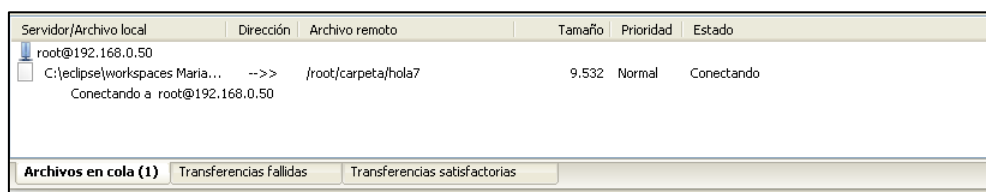


Ilustración 115: Detalle 2 transferencia de archivos

### 15.4.1 Problemas en la comunicación

Las posibles soluciones a los problemas que pueden producirse durante la comunicación son:

- Comprobar si la placa TS-7200 está correctamente alimentada.
- Comprobar si el cable Ethernet está conectado al PC y a la placa.
- Problema en la configuración: Repetir el proceso de configuración del *sitio* (punto 3 del apartado 15.3).
- Repetir el proceso de transferencia.
- Consultar a las pestañas: **Trasferencias fallidas** y **Trasferencias satisfactorias**.

### 15.4.2 Finalización de la comunicación mediante FileZilla

La comunicación mediante el programa FileZilla se cierra en *Archivo* → *Salir*.

Nota: Finalizar la comunicación mediante el FileZilla no implica que la tarjeta este apagada. Para apagar la tarjeta correctamente se debe escribir el comando: ***shutdown -h now*** y proceder a suprimir la alimentación.



## 16 COMUNICACIÓN

La tarjeta TS-7200 no dispone de interfaz gráfica con el usuario, así pues, los datos se presentarán mediante un *programa usuario* instalado en los equipos del sistema en los que sea necesario una gestión de la batería. Los datos suministrados por la tarjeta controladora serán transferidos al programa usuario mediante *sockets* a través de la red Ethernet. Un *socket* es un interfaz normalizado de comunicación (API *Aplication Programming Interface*) entre procesos, en este caso ejecutado en una red TCP/IP (*Transmission Control Protocol / Protocol Internet*) [109]. La tarjeta y los equipos, se comunican mediante su dirección IP y un número de puerto. La arquitectura TCP/IP establece que sean los clientes (equipos alimentados) los que realicen las peticiones al servidor (tarjeta TS-7200) y que este último responda a las peticiones de los clientes.

### 16.1 INTRODUCCIÓN A LOS SOCKETS

Los *sockets* permiten la comunicación entre procesos que son ejecutados en diferentes máquinas. Cada *socket* se identifica mediante una dirección IP y un número de puerto. La transmisión de datos entre la tarjeta TS-7200 y los equipos se ha realizado en arquitectura TCP/IP garantizando una transmisión de datos correcta y completa. La Ilustración 116 refleja la secuencia de eventos que se producen mediante la transmisión de datos de la tarjeta controladora y los equipos en arquitectura TCP/IP, donde el proceso cliente inicia la comunicación mientras que el proceso servidor está a la espera de que la conexión sea iniciada, para comenzar la comunicación. Si alguno de los procesos permanece ocupado y no atiende a la comunicación el otro proceso permanecerá a la espera hasta conseguir recibir o enviar los datos.

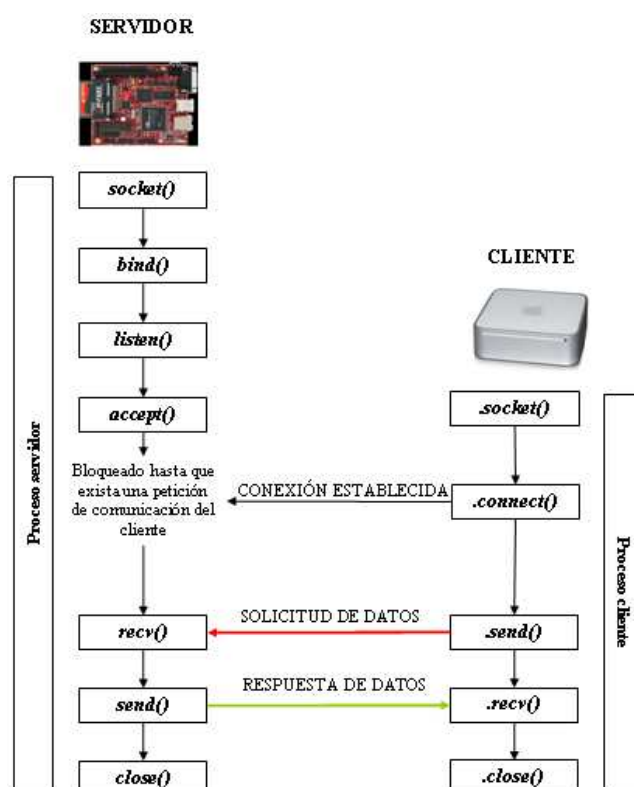


Ilustración 116: Diagrama de bloques en la comunicación TCP/IP

El protocolo TCP/IP se utiliza cuando es importante que no se pierdan datos durante la transferencia de información y el tiempo de respuesta de la información no es crítico.

## 16.2 SOCKET EN C

### 16.2.1 Creación de un socket

La Ilustración 116 representa el diagrama de bloques de la comunicación TCP/IP entre la tarjeta TS-7200 (proceso servidor, escrito en lenguaje C) y los equipos (proceso cliente, escrito en lenguaje Python), en ella se puede observar como la comunicación se inicia con la creación de un *socket*. Los *sockets* se crean llamando a la función *socket()* que devuelve un entero que corresponde al identificador de socket (en el programa corresponde a la variable *sock*). Si se produce un error durante la creación del socket, la función devuelve -1 y se recibe el aviso por pantalla de “*Error socket*”. La función *socket* requiere un dominio donde realizar la conexión, el tipo de sockets y el protocolo a utilizar. El identificador *sock* se ha creado dentro del dominio AF\_INET que es el que utilizan los procesos que se comunican a través de una red TCP/IP. En el dominio AF\_INET se definen los sockets *Stream* cuando se utiliza el protocolo TCP/IP que proporciona un flujo de datos orientado a la conexión, bidireccional sin duplicación de paquetes y libre de errores.

Para enviar los datos a través de la red conveniente convertir los datos del formato de máquina al formato de red y del formato de red al formato de máquina al recibirlos, las conversiones se realizan mediante las funciones *htons()* y *ntohs()*. En la Ilustración 117 las flechas salientes de las máquinas representan el envío de datos y las flechas entrantes la recepción de datos.

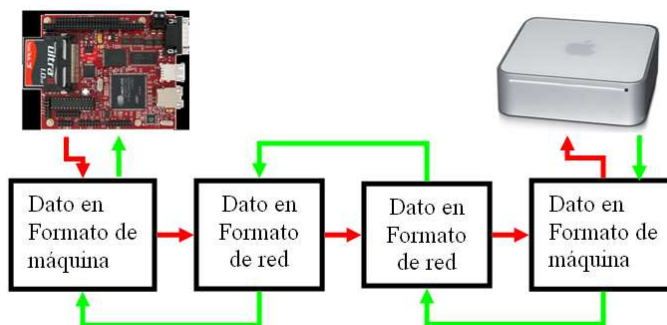


Ilustración 117: Cambio de formato en la transmisión de datos

### 16.2.2 Función *bind()*

La función *bind()* define el puerto al que van a conectarse los clientes. El número de puerto seleccionado deberá estar comprendido entre un valor mayor que 1024 hasta 65535. En caso en el que se haya producido alguna situación de error, mostrará por pantalla el mensaje: “*Error en bind*”.

La función *bind()* está definida por tres argumentos el primero toma el valor que proporciona el identificador del socket tras la llamada a la función *socket()*, el segundo es un puntero a una estructura y el tercero proporciona el tamaño en bytes de la estructura anterior mediante el operador *sizeof()*. La estructura más interesante para las redes TCP/IP es: *sockaddr\_in* que es equivalente a la *sockaddr* pero que permite referenciar a los elementos de forma más sencilla que esta última. Los campos de la estructura *sockaddr\_in* son: *.sin\_family*, *.sin\_port*, *.sin\_addr*. En el campo *.sin\_family* se define el protocolo; Los procesos que se



comunican a través de una red TCP/IP requieren asignar en este campo el dominio comunicación: *AF\_INET*. El campo *.sin\_port* representa el número de puerto. Se ha programado que los paquetes se reciban por el puerto 5000 [111], este puerto es el dispositivo universal de *plug and Play* (UPnP) el cual permite el intercambio de información y datos entre los dispositivos conectados a una red. Por último el campo *.sin\_addr* se almacena la dirección IP; Asignando a este campo la constante *INADDR\_ANY* se realizará la asignación de la dirección IP de forma automática, sin tener por que conocer previamente la dirección IP donde se va a ejecutar el programa. Esto le indica al sistema que el programa recibirá mensajes por cualquier IP válida de la máquina, en el caso de disponer de varias.

### 16.2.3 Función *listen()* y Función *accept()*

El servidor invoca a la función *listen()* y habilita el socket para poder recibir conexiones, en caso de error se recibe por pantalla el mensaje “*Error listen*”. Si la habilitación ha sido correcta aparecerá por pantalla el mensaje: “*Servidor TCP esperando a cliente por puerto 5000*”. Después se ejecuta la función *accept()* y permanece a la espera hasta que algún cliente establezca conexión con el servidor, finalizando en el momento en el que se produzca la conexión. Al conectarse la función *accept()* devuelve un parámetro que se convierte en el nuevo identificador del socket (necesario para realizar la transferencia de datos por la red), dejando así libre el socket para poder atender nuevas peticiones. Iniciada la conexión el mensaje por pantalla indica desde que puerto y dirección IP se consiguió la conexión.

### 16.2.4 Enviar y recibir datos

Una vez establecida la comunicación se utilizan las funciones *send()* y *recv()* para realizar la transferencia de datos. La función *send()* devuelve el número de bytes enviados mientras que la función *recv()* se utiliza para recibir datos devolviendo el número de bytes y no finaliza hasta que recibe algún tipo de información o los caracteres “q” o “Q”. El primer dato que envía la tarjeta TS-7200 al conectarse es un mensaje de bienvenida del tipo: “*Bienvenido a la placa TS-7200*”. Los datos recibidos son mostrados por pantalla precedidos del mensaje: “*DATO RECIBIDO =*” y los datos enviados son precedidos del mensaje: “*DATO ENVIADO =*”

### 16.2.5 Finalizar la conexión

Para finalizar la conexión se utiliza la función *close()*, una vez utilizada, deshabilita el socket no permitiendo realizar transferencias de datos (ni envíos ni recepciones).

## 16.3 SOCKET EN PYTHON

El proceso cliente escrito en Python es ejecutado en los equipos. Importado el módulo socket, la función *MyDialogSetCommunication* se encarga de establecer la comunicación con la tarjeta controladora. Al invocar al método *connect* se conectarse con la tarjeta pasando como parámetros el “localhost” o la dirección IP en este caso 192.168.0.50 [105] y el puerto 5000. Para recibir datos se invoca al método *recv()* pasándole como parámetro el número máximo de bytes que aceptará la comunicación, en este caso 512. Para enviar peticiones de información al

proceso servidor se invoca el método *send()* pasando como parámetro una cadena de caracteres. La comunicación finaliza cuando el usuario pulsa el botón *Cerrar conexión*, en ese momento se envía el carácter “q” que pone fin a la conexión con el servidor.

## 16.4 SECUENCIA DE EVENTOS EN LA COMUNICACIÓN

Al ejecutar el programa servidor aparecen en la consola donde se está ejecutando dos mensajes, el primero es un mensaje de bienvenida a la tarjeta y otro que indica que el servidor está disponible y el puerto por el cual recibe peticiones tal y como muestra la Ilustración 118.

PROGRAMA EasyBattery.py (Cliente)		
Función	Funcionalidad	Solicitud
<b>MyDialogSetCommunication</b>	Establece comunicación	0
<b>MyDialogCorriente</b>	Proporciona información sobre la corriente de consumo la capacidad actual y el tiempo de descarga	6
<b>MyDialogRed</b>	Proporciona información sobre si el sistema está conectado o no a la red.	2
<b>MyOnCapacity</b>	Proporciona información sobre la capacidad estimada de la batería	5

Tabla 35: Tabla de solicitudes del programa cliente

Bienvenido a la placa TS-7200  
Servidor TCP esperando cliente por puerto 5000

Ilustración 118: Mensajes en la consola al inicio de la comunicación

Una vez establecida la comunicación el programa servidor permanece a la espera de la solicitud de datos del programa cliente. En la Tabla 35 aparecen las funciones del programa cliente que realizan solicitudes al programa servidor. Las solicitudes son los datos que son transferidos a través del socket que realizan una petición de información al servidor. El servidor al recibir la solicitud devuelve la información correspondiente. La relación de la información devuelta por el servidor en función de la solicitud aparece en la Tabla 36. Se han implementado solicitudes que en la actualidad no son requeridas por el programa cliente, pero que serán de gran utilidad en futuras aplicaciones.

PROGRAMA EasyBattery.c (Servidor)	
Solicitud	Respuesta
0	Si se recibe comunicación y no hay una solicitud se envía la solicitud 0 por defecto
6	Devuelve la corriente que demandan las cargas a la batería.
2	Devuelve si está conectado o no a la red.
5	Devuelve el porcentaje de capacidad estimada de la batería.
3	Activa la salida
4	Desactiva la salida
1	Devuelve la tensión de entrada en el ADC0
Cualquier otro valor	Indica por pantalla que el código introducido es erróneo.

Tabla 36: Tabla de funcionalidades del programa servidor en función de la solicitud requerida por el programa cliente

## 17 EASYBATTERY

Como se comentó, la tarjeta controladora TS-7200 no tiene interfaz de usuario, por esta razón se ha desarrollado un interfaz gráfico que estará disponible en los equipos que alimenta el sistema. La comunicación entre la tarjeta y los equipos se realiza mediante *sockets* en arquitectura TCP/IP, permitiendo que los procesos puedan recibir y transmitir datos a través de la red Ethernet. En resumen, la tarjeta, de dirección IP 192.168.0.50 [105], actúa como servidor remoto, respondiendo a las peticiones del programa cliente que ha sido desarrollado en wxPython con el nombre de *EasyBattery*.

### 17.1 ACCESO AL PROGRAMA EASYBATTERY

Haciendo doble click sobre el ejecutable *EasyBattery.py* que se encuentra disponible en la carpeta *EasyBattery* (Ilustración 119), esta carpeta incluye todos los archivos necesarios para poder ejecutar correctamente el programa.

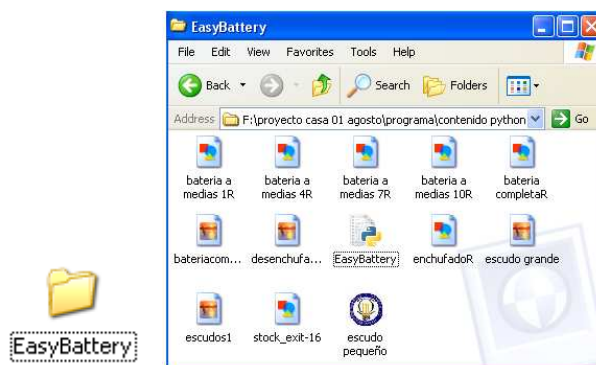


Ilustración 119: Carpeta EasyBattery

### 17.2 ENTORNO

En la Ilustración 120 aparece el entorno de trabajo de *EasyBattery*. Para poder gestionar el sistema se debe proceder en primer lugar a conectar con la tarjeta controladora TS-7200, esto se realiza presionando el botón remarcado en la parte izquierda de la Ilustración 120 con el nombre de “*Establecer comunicación*”.

#### 17.2.1 Establecer conexión

En el caso de que la comunicación no se haya establecido, al pulsar “*Comprobar Conexión de red*” o “*Comprobar intensidad demanda*” que aparecen señaladas con una flecha en

la Ilustración 120 mostrarán los mensajes de las ilustraciones 120 y 121 respectivamente, indicando al usuario que la conexión no se ha iniciado.



Ilustración 120: Entorno de EasyBattery

La conexión puede no haberse iniciado por varios motivos:

- ✓ No haber pulsado el botón “*Establecer comunicación*”.
- ✓ No estar conectada la batería del vehículo, ya que es esta batería la que alimenta la tarjeta TS-7200.
- ✓ No hay conexión con la tarjeta TS-7200: indica al usuario que el servidor está desconectado: La tarjeta TS-7200 está funcionando perfectamente pero el ejecutable instalado en la tarjeta TS-7200 con el nombre de EasyBattery, no se está siendo ejecutado en ese momento.

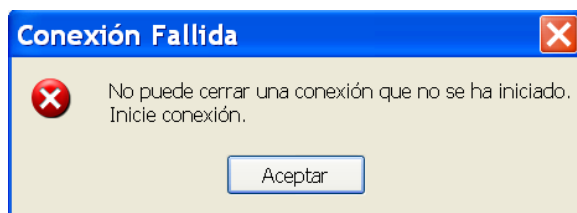


Ilustración 121: Mensaje error de conexión al pulsar el botón “*comprobar Conexión de red*”



Ilustración 122: Mensaje error de conexión al pulsar el botón “*Comprobar intensidad demanda*”

Si no se ha establecido conexión y se pulsa el botón “*Cerrar Comunicación*” (señalado con una flecha en la Ilustración 120, el usuario recibirá un mensaje de error como el de la Ilustración 123, que avisa al usuario que no puede realizar una desconexión si previamente no se ha realizado la conexión.

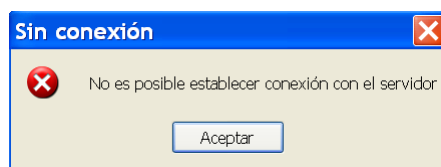


**Ilustración 123:** Mensaje de error de conexión al pulsar el botón “Cerrar Comunicación”

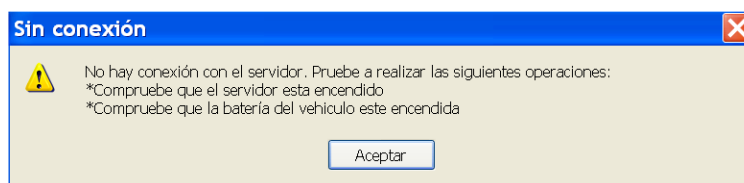


**Ilustración 124:** Mensaje de conexión

Una vez se ha establecida la comunicación aparece el mensaje de la Ilustración 124 indicando al usuario que se ha establecido la conexión con el servidor (la tarjeta TS-7200). Si se pulsa el botón *Establecer comunicación* y no hay conexión con el servidor, aparecerá un mensaje como el de la Ilustración 125 comunicándole al usuario que no ha sido posible establecer comunicación. Si el cable de Ethernet que comunica la tarjeta TS-7200 con el equipo cliente no está conectado, este mensaje tardará unos segundos en aparecer si por el contrario está conectado será instantáneo. Al pulsar **Aceptar** aparecerá un mensaje aviso como el de la Ilustración 126 con recomendaciones al usuario para restablecer la comunicación.



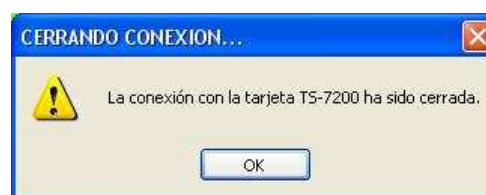
**Ilustración 125:** Mensaje error sin conexión



**Ilustración 126:** Mensaje aviso sin conexión

## 17.2.2 Cerrar comunicación

Una vez establecida la comunicación para cerrarla existen dos posibilidades volver a presionar el botón *Establecer comunicación* o pulsar el botón *Cerrar comunicación* que fuerza a la comunicación a cerrarse.



**Ilustración 127:** Mensaje Cerrar Comunicación

## 17.3 BOTÓN COMPROBAR CONEXIÓN A LA RED

Una vez establecida la conexión al servidor, si se pulsa al botón: “Comprobar Conexión de red” en la ventana principal, si el sistema está conectado a la red, la ventana “Comprobando conexión a la red” será como muestra la Ilustración 128 y si el sistema no está conectado, esta misma ventana mostrará la Ilustración 129.





Ilustración 128: Ventana aviso del botón: “Comprobar Conexión de red” que indica que el Sistema no está conectado a la red.



Ilustración 129: Ventana aviso del botón: “Comprobar Conexión de red” que indica que el Sistema está conectado a la red.

## 17.4 BOTÓN COMPROBAR INTENSIDAD DEMANDADA

Siempre y cuando la conexión esté establecida, pulsando en la ventana principal el botón: “Comprobar intensidad demanda” se accede a la ventana llamada “Comprobando intensidad demandada” similar a la Ilustración 130, donde aparece información relativa a la capacidad actual de la batería adicional que incorpora el vehículo, el consumo de corriente demandado por las cargas y el tiempo de duración de la batería en horas. Toda la información que aporta esta ventana está referida a la capacidad nominal de la batería que por defecto es de 90Ah.



Ilustración 130: Ventana "Comprobando intensidad demandada"

### 17.4.1 Modificación de la capacidad de la batería actual

Si se desea cambiar el valor nominal de la batería adicional por una batería de capacidad diferente a 90Ah, se debe ir a **Opciones** en el menú principal. Seleccionando: **Opciones->Batería adicional** o mediante el acceso directo Ctrl+B, se accederá a la ventana: **Selección capacidad** que parece en la Ilustración 131. Mediante el desplegable, se puede introducir el nuevo valor de capacidad de la batería, una vez pulsado el botón **Guardar** el valor se habrá modificado.

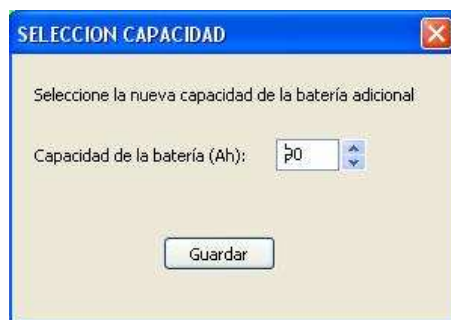


Ilustración 131: Ventana: Selección de la capacidad

### 17.4.2 Tabla de datos

Los datos que aparecen en la ventana "*Comprobando intensidad demandada*" pueden verse reflejados en una tabla. Para acceder a la tabla denominada "*Visualización de datos*", se selecciona **Datos** en el menú principal: **Datos->Datos** o mediante el acceso directo Ctrl+D. Si hay conexión con el servidor la tabla mostrará la corriente nominal de consumo (Cc) en Amperios, el tiempo de descarga Tc (tiempo que tardaría la batería en descargarse con el consumo actual y la actual carga de la batería) y el porcentaje de carga de la batería en ese momento, la hora y la fecha en la que se ha realizado la última consulta, tal y como aparece en la Ilustración 132. Si *Easybattery* no estuviera conectado con la tarjeta TS-7200, los datos relativos a la corriente y al tiempo aparecería un mensaje indicando que no hay conexión con la tarjeta TS-7200.



Ilustración 132: Visualización de datos cuando hay conexión con la tarjeta TS-7200

### 17.4.3 Eliminar

Cada vez que se accede a **Datos**, la información reflejada en la tabla de *Visualización de datos*, aparece reflejada en la pestaña *Impresión de Datos* de la ventana principal (ver Ilustración 133). Siempre y cuando se haya comprobado previamente la intensidad demandada. Si en alguna de las columnas aparecen las siglas SA significan *Sin Actualización* que el valor no está actualizado, y se debe pulsar el botón *Comprobar intensidad demandada*. Las siglas SC indican *Sin Conexión* que no está conectado con el servidor (la tarjeta TS-7200), así como las siglas SB indican *Sin Batería*. La pestaña *Impresión de Datos* no puede ser modificada para evitar la manipulación de los datos. La información puede ser eliminada mediante: **Datos->Eliminar**, esta acción hace que la pestaña *Impresión de Datos* quede en blanco.

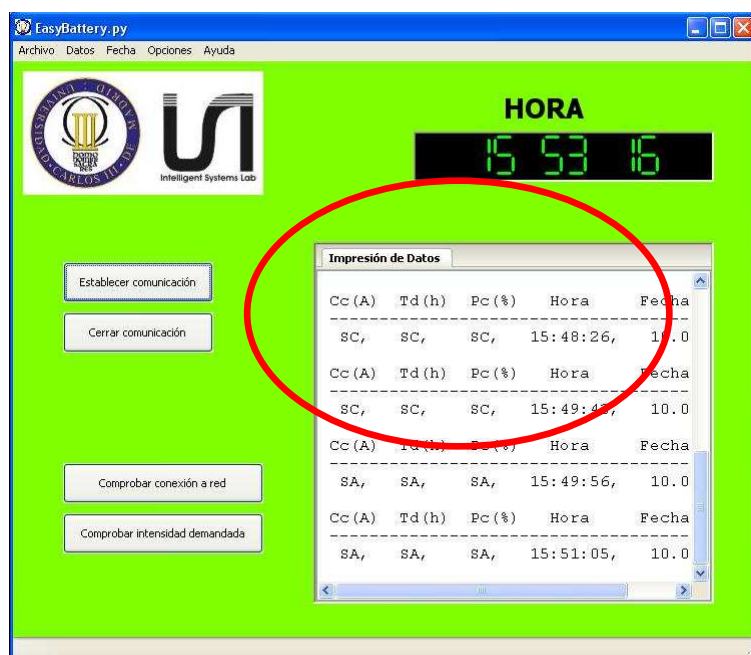


Ilustración 133: Eliminar datos de la Pestaña Impresión de Datos

## 17.5 PESTAÑA IMPRESIÓN DE DATOS

### 17.5.1 Guardar un documento

La pestaña *Impresión de Datos* puede guardarse como archivo de texto para poder trabajar con los datos, por ejemplo, en una aplicación Excel. Los datos se guardan en la ventana “**Guardar como:**” (Ilustración 134) para acceder a esta ventana, se debe ir a **Archivo**, del menú principal: **Archivo->Guardar** o mediante el acceso directo Ctrl+S. En el recuadro **Nombre** se debe incluir el nombre del documento terminado en .txt y una vez seleccionado el directorio donde se desea guardar el documento, pulsar el botón **Guardar**.

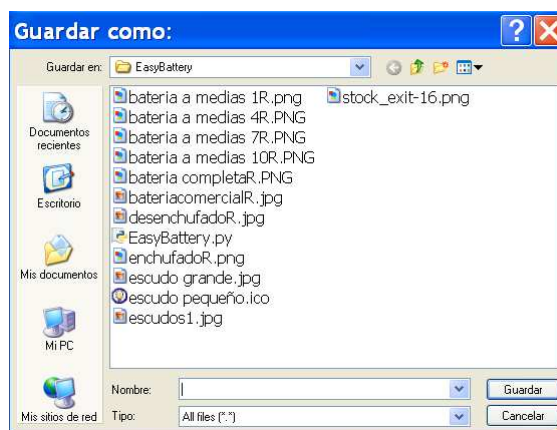


Ilustración 134: Ventana Guardar como

### 17.5.2 Vista previa de un documento

Los datos de la pestaña de *Impresión de Datos* pueden ser visualizados previamente antes de ser guardados o impresos. La visualización previa parece en: **Archivo->Vista previa**. El documento se puede ampliar o disminuir mediante el desplegable situado a la derecha de la ventana (ver Ilustración 135).

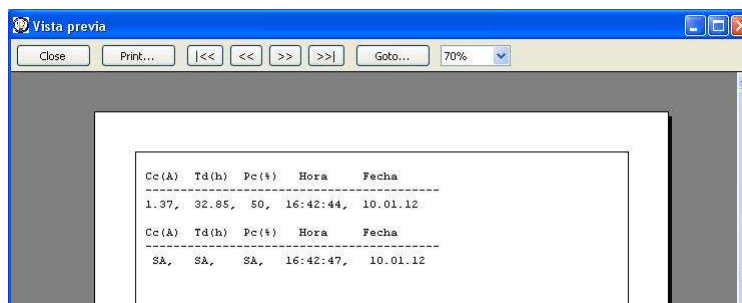


Ilustración 135: Ventana Vista previa



Desde esta ventana se puede imprimir el documento pulsando el botón **Print**. Las comillas se utilizan para desplazarse entre las diferentes hojas en el caso en el que el documento tenga más de una. Para salir de **Vista Previa** pulsar el botón **Close**.

### 17.5.3 Configurar un documento

La configuración de un documento se realiza desde “**Configurar página...**” en **Archivo** del menú principal. En la ventana **Configurar página** que aparece en la Ilustración 136, se puede cambiar la orientación del documento en el recuadro **Orientación**, el tamaño en el recuadro **Papel** o los márgenes en el recuadro **Márgenes**.

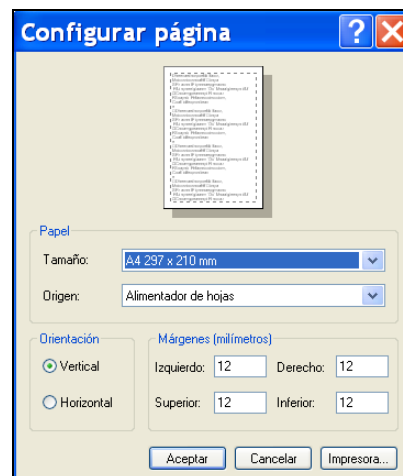


Ilustración 136: Ventana Configurar página

### 17.5.4 Imprimir un documento

La impresión de un documento se realiza desde “**Imprimir**” en **Archivo** del menú principal o mediante el acceso directo Ctrl+P. La ventana imprimir puede verse en la Ilustración 137. El documento se puede imprimir mediante impresora o si se dispone de un creador de PDFs en archivo PDF. Para imprimir pulsar Print.

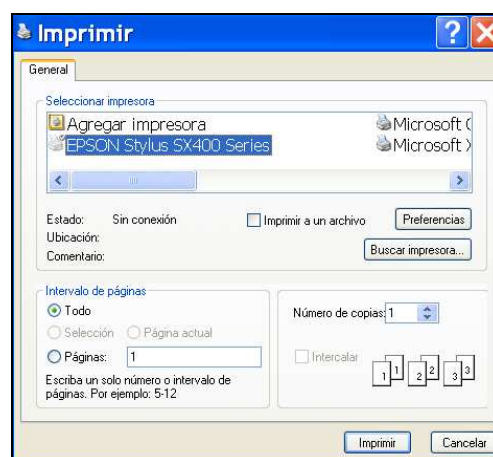


Ilustración 137: Ventana Imprimir

## 17.6 CAPACIDAD ESTIMADA

En el menú **Datos** también se puede encontrar la opción **Capacidad estimada** (ver Ilustración 138), que indica al usuario el porcentaje de carga en la batería. El porcentaje de carga podrá tener un valor comprendido entre 0 y 100% tal y como puede verse en la Ilustración 139, este valor se obtiene a partir de medir la tensión en bornes de la batería de acuerdo con el apartado 4.4.3. Cuando la capacidad estimada es del 0% se pide al usuario que recargue la batería cuanto antes.



Ilustración 138: Acceso a Capacidad estimada



Ilustración 139: Porcentajes de la capacidad de la batería



Ilustración 140: Mensaje error de conexión al pulsar el botón “Capacidad estimada”

En el caso de se acceda a la opción “*Capacidad estimada*” sin haber establecido previamente conexión con la tarjeta TS-7200, aparecerá el mensaje como el de la Ilustración 140, el cual es similar al de las ilustraciones 114 y 115.

## 17.7 CONFIGURACIÓN DE LA VENTANA PRINCIPAL

### 17.7.1 Fondo

El color de fondo de la ventana principal de *EasyBattery* es configurable por el usuario, permite poner colores oscuros en condiciones de conducción nocturna. La modificación se realiza desde la pestaña *Opciones* del menú principal: **Opciones** → **Fondo** o mediante el acceso directo Ctrl+F. Una vez seleccionado el color en la ventana **Color** que aparece en la Ilustración 141, se pulsa el botón *Aceptar* y el fondo de pantalla cambia tal y como aparece en la Ilustración 142.

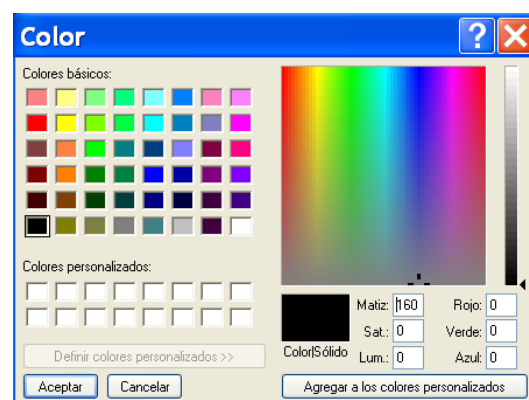


Ilustración 141. Ventana Color

Una vez cerrado el programa se mantiene la configuración predeterminada.

## 17.7.2 Fecha y hora

Dentro de las configuraciones de la ventana principal se puede añadir la fecha actual. Si se accede a “**Fecha**” desde **Fecha** del menú principal aparecerá la fecha actual debajo del display de **HORA**, tal y como aparece en la Ilustración 142. La Hora aparece en la ventana principal del programa por defecto y no es configurable. Tanto la fecha como la hora están sincronizadas con la hora del sistema en el que se esté ejecutando el cliente, para cambiarlas hay que cambiar la fecha y la hora del sistema.

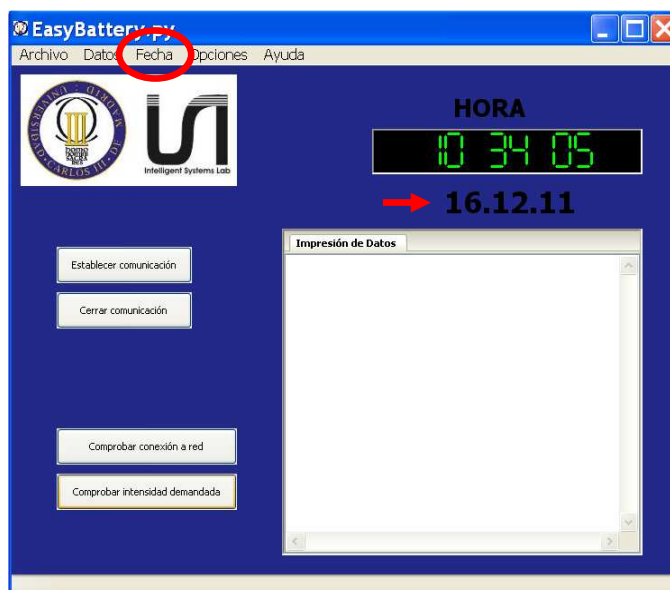


Ilustración 142: Añadir la *Fecha* a la ventana principal

## 17.8 AYUDA

### 17.8.1 Acerca de Easybattery

La ventana Acerca de *EasyBattery* (Ilustración 143) contiene información sobre el programa *EasyBattery*, versión, desarrollador, correo electrónico de contacto y un enlace con la página del laboratorio de Sistemas de Sistemas Inteligentes. Se accede a esta ventana mediante: **Ayuda->Acerca de Easybattery**. Para salir se debe pulsar el botón **OK**.



Ilustración 143: Ventana Acerca de *EasyBattery*



# 18 PRESUPUESTO

## 18.1 DESGLOSE DE COSTES

En la Tabla 37 aparece un desglose del presupuesto del sensor de tensión, del sensor de corriente y el presupuesto final para el conjunto del sistema.

DESGLOSE SENSOR DE TENSION		
COMPONENTES	UNIDADES	PRECIO
TRANSFORMADOR TR4106	1	2,97 €
GBPC2506	1	3,21 €
RESISTENCIAS varias	3	0,5 €
CONDENSADORES 100n	2	0,2 €
LM324	1	0,61 €
Zener BZX85C5V6	1	0,18 €
Zener BZX85C4V3	1	0,18 €
Zener BZX85C3V3	1	0,19 €
Varios	1	0,3 €
TOTAL:		9,54 €

DESGLOSE SENSOR DE CORRIENTE		
COMPONENTES	UNIDADES	PRECIO
REISTENCIAS varias	3	0,2 €
LM324	1	0,61 €
CONDENSADORES 0,15u	2	0,3 €
zener BZX85C4V3	1	0,18 €
Varios	1	0,3 €
TOTAL:		2,29 €

COSTE DE PERSONAL		
PERSONAL	HORAS	PRECIO
INGENIERO TÉCNICO INDUSTRIAL: ELÉCTRÓNICO	1763	10,6 €
TOTAL:		18687,8 €

PRESUPUESTO FINAL		
COMPONENTES	UNIDADES	PRECIO
SENSOR DE TENSIÓN	1	9,54 €
SENSOR DE CORRIENTE	1	2,29 €
TS-7200	1	134 €
SAI APC/700	1	93,47 €
BATERÍA	1	142,64 €
INVERSOR	1	104,91 €
PERSONAL	1	18687,80 €
TOTAL:		19174,65 €

Tabla 37: Presupuesto<sup>20</sup>

El presupuesto final en materiales asciende a **diecinueve mil ciento setenta y cuatro con sesenta y cinco céntimos**.

<sup>20</sup> Precios válidos a 09/12/11



## 19 TRABAJOS FUTUROS

Dentro de los trabajos futuros trabajos se pueden distinguir:

- **Fabricar los prototipos de los sensores.-** En la actualidad los prototipos están soportados sobre una protoboard, queda pendiente materializar el diseño OrCAD.
- **Industrializar los prototipos.-** Revisar el diseño para hacer posible la fabricación a gran escala. La industrialización del producto debe tener en cuenta que este debe soportar el funcionamiento normal del vehículo: aceleraciones, pequeños golpes...
- **Integrar en el vehículo los sensores y placa controladora.-** Buscar una ubicación dentro de la estructura del vehículo a resguardo de polvo, humedad, focos de calor y accesible, mantenible dentro las operaciones normales de mantenimiento y de reparación.

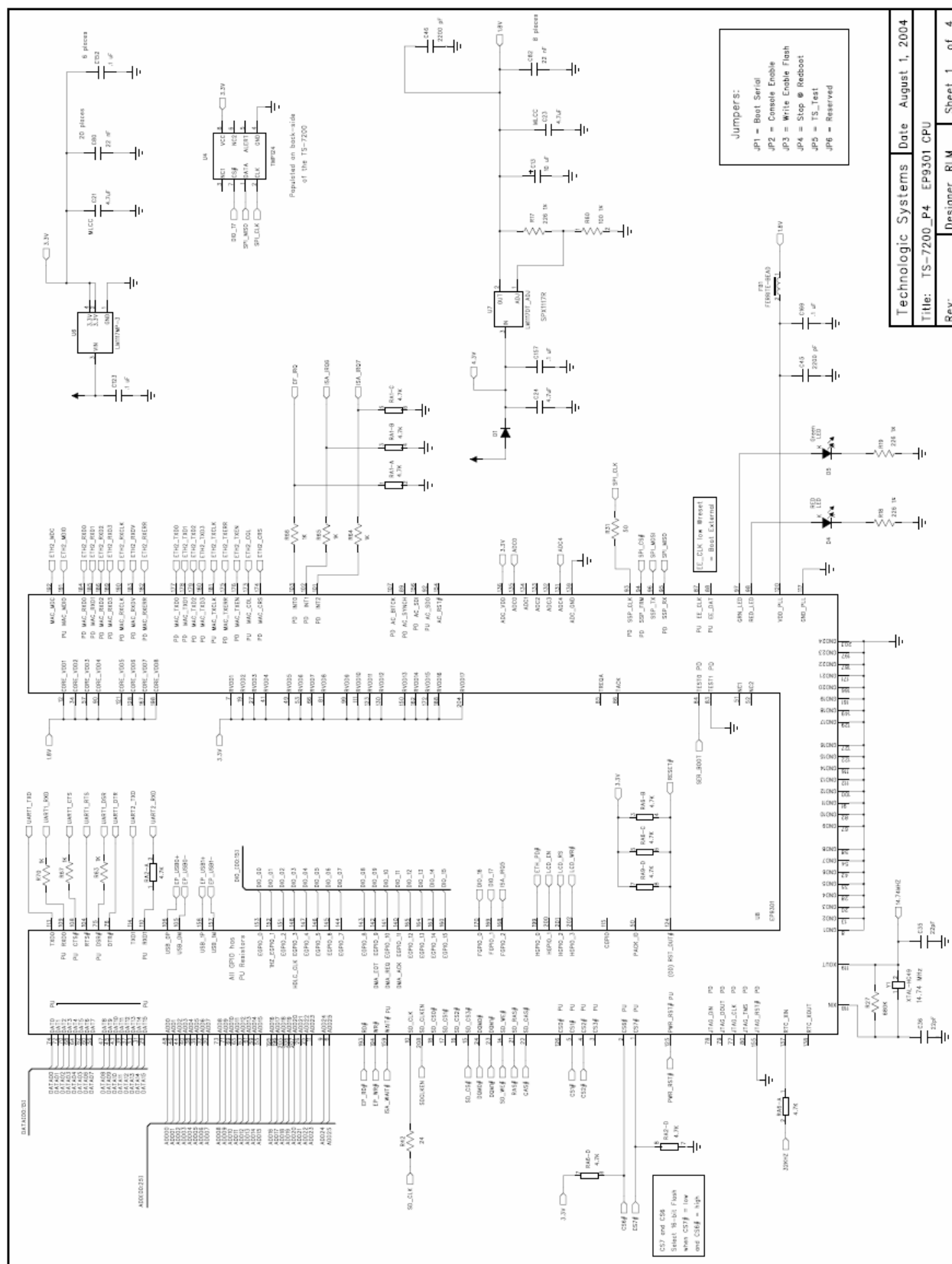


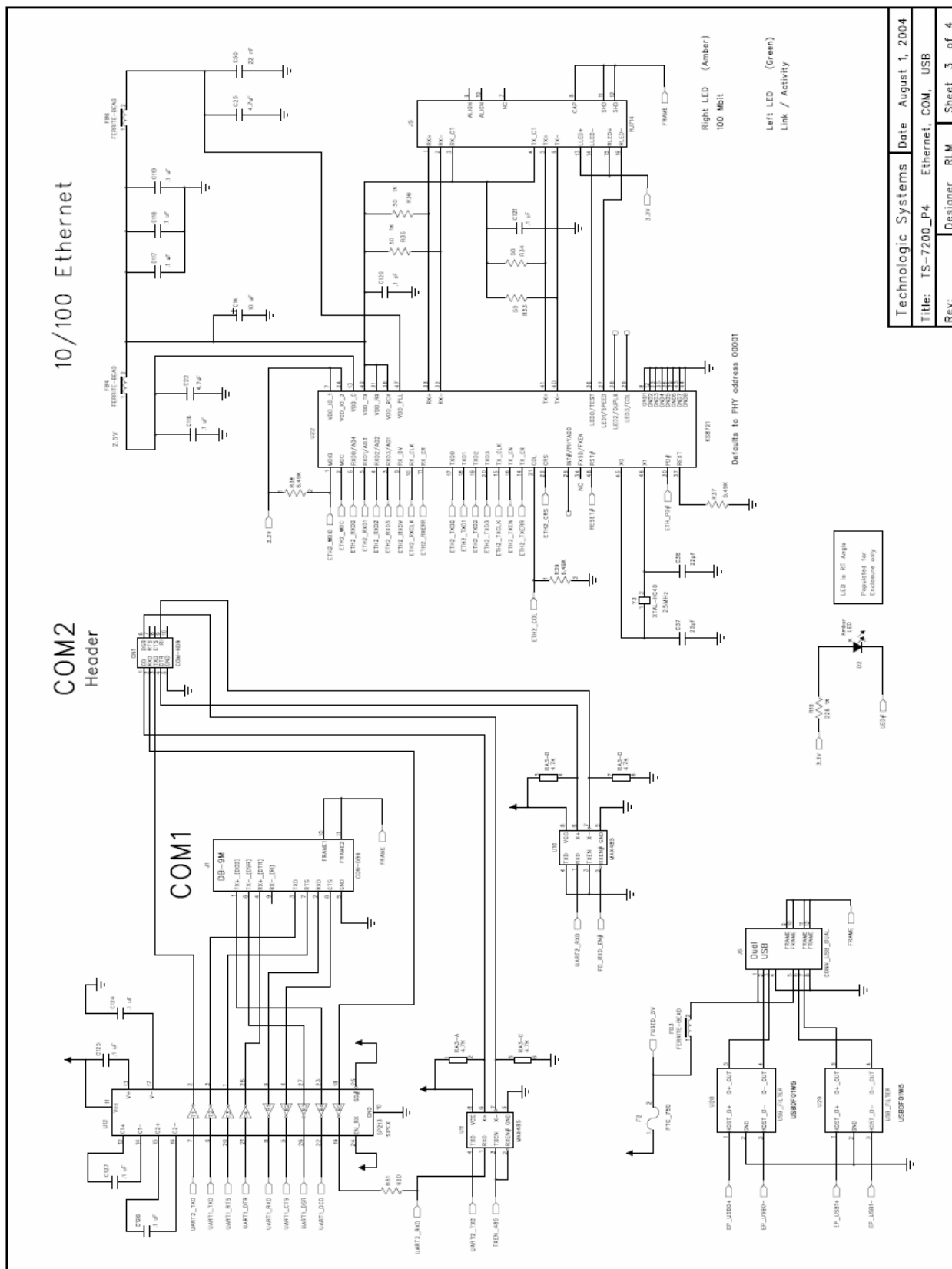


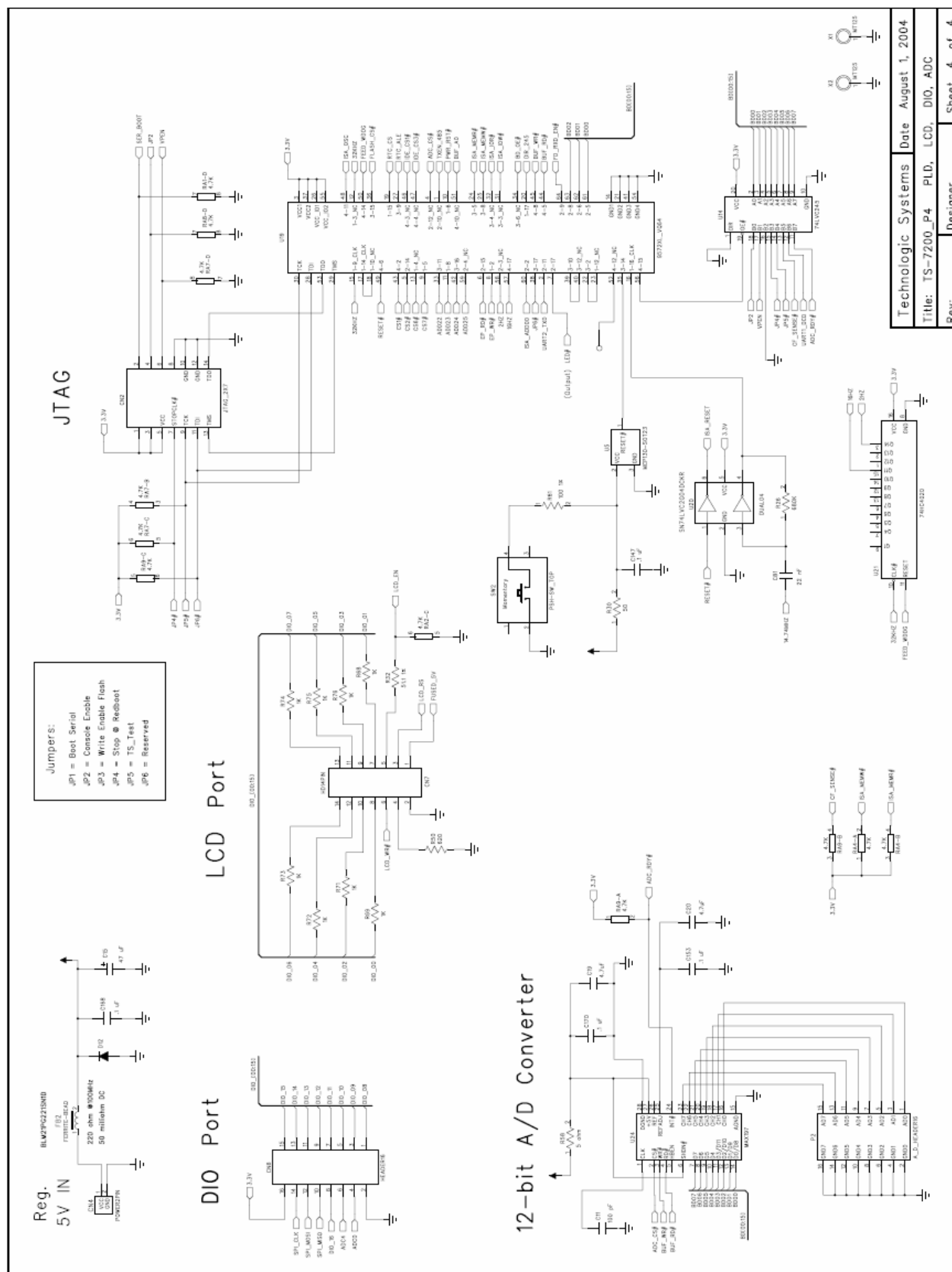
## **20 APÉNDICES**



Fuente: Technologic Systems







## 20.2 APENDICE B: Mapa de memoria

Address Region	Function
0xF000_0000 - 0xFFFF_FFFF	nCS0 (not used)
0xE000_0000 - 0xEFFF_FFFF	SDRAM (TS-7250)
0xD000_0000 - 0xDFFF_FFFF	SDRAM (not used)
0xC000_0000 - 0xCFFF_FFFF	SDRAM (not used)
0x8084_0000 - 0x8084_00C8	GPIO control registers
0x8000_0000 - 0x800F_FFFF	AHB mapped registers
0x71C0_0000 - 0x71FF_FFFF	CS7 PC/104 8/16 bit I/O (user selectable timing)
0x7180_0000 - 0x71BF_FFFF	CS7 PC/104 8/16 bit Memory (selectable timing)
0x7000_0000 - 0x70FF_FFFF	TS-9420 Flash
0x7000_0000 - 0x7FFF_FFFF	CS7 (bit bus cycles)
0x6000_0000 - 0x60FF_FFFF	on-board Flash (TS-7200)
0x6000_0000 - 0x6FFF_FFFF	CS6 (Flash)
0x3000_0000 - 0x3FFF_FFFF	CS3 (not used)
0x21C0_0000 - 0x21FF_FFFE	PC/104 16-bit I/O
0x2180_0000 - 0x21BF_FFFF	PC/104 16-bit Memory
0x2000_0000 - 0x2FFF_FFFF	CS2 (16-bit bus cycles)
0x11C0_0000 - 0x11FF_FFFF	PC/104 8-bit I/O
0x1180_0000 - 0x11BF_FFFF	PC/104 8-bit Memory
0x1000_0000 - 0x1FFF_FFFF	CS1 (8-bit bus cycles)
0x0001_0000 - 0x0000_FFFF	SDRAM region

Register Address	Function
0x8090_0020	Cirrus A/D lock register
0x8090_0018	Cirrus A/D channel select register
0x8090_0008	Cirrus A/D result register (RO)
0x808D_0000 - 0x808D_FFFF	UART2 control registers
0x808C_0000 - 0x808C_FFFF	UART1 control registers
0x808A_0000 - 0x808A_FFFF	SPI control registers
0x8084_0044	LCD_EN, LCD_RS, LCD_WR direction reg.(bits3-5)
0x8084_0040	LCD_EN, LCD_RS, LCD_WR data reg.(bits3-5)
0x8084_0034	DIO_8 direction register (bit 1)
0x8084_0030	DIO_8 data register (bit 1)
0x8084_0020	On-board LEDs register (bits 0, 1)
0x8084_0018	Port C direction register (TS-7300) LCD_7 direction register (bit 7)
0x8084_0014	DIO_0 thru DIO_7 direction register (R/W)
0x8084_0010	LCD_0 thru LCD_7 direction register (R/W)
0x8084_0008	Port C data register (TS-7250) 1000 mA driver output on DIO (bit 0) (TS-7300) LCD_7 data register (bit 7)
0x8084_0004	DIO_0 thru DIO_7 data register (R/W)
0x8084_0000	LCD_0 thru LCD_7 data register (R/W)
0x8081_0000 - 0x8081_FFFF	Timer Control registers
0x8080_0000 - 0x8FFF_FFFF	APB mapped registers
0x800B_0000 - 0x800B_FFFF	VIC 0 registers
0x8006_0000 - 0x8006_FFFF	SDRAM control registers
0x8002_0000 - 0x8002_FFFF	USB registers
0x8001_0000 - 0x8001_FFFF	Ethernet MAC registers
0x7220_0000 - 0x729F_FFFF	(TS-7300) FPGA SDRAM region

Register Address	Function
0x7210_0000 - 0x7200_FFFF	(TS-7300) FPGA MAC core 32-bit registers
0x7200_0044 - 0x7200_0047	(TS-7300) FPGA DIO2 TS-XDIO #2 registers
0x7200_0040 - 0x7200_0043	(TS-7300) FPGA DIO2 TS-XDIO #1 registers
0x7200_0030 - 0x7200_003A	(TS-7300) FPGA Video core 16-bit registers
0x7200_0020 - 0x7200_0027	(TS-7300) FPGA SD Card core 16-bit registers
0x7200_0000 - 0x7200_001F	(TS-7300) FPGA COM3-4-5-6-7-8 16-bit registers
0x6000_0000	(TS-7250) NAND Flash data register
0x6040_0000	(TS-7250) NAND Flash control register (bits 0-2)
0x6080_0000	(TS-7250) NAND Flash Busy status (bit 5)
0x23C0_0000	WDT Feed register (bits 0-2)
0x2380_0000	WDT Control register (bits 0-2)
0x2340_0000	PLD version (bits 0-2)
0x2300_0000	COM2 RS-485 control register
0x22C0_0000	COM2 RS-485 control register (bits 0-2)
0x2280_0000	JP6 (bit 0)
0x2280_0000	Bootling from TS-9420 (bit 1)
0x2280_0000	TS-9420 present (bit 2)
0x2240_0000	MAX197 A/D option present (bit 0)
0x2240_0000	COM2 RS-485 option present (bit 1)
0x2200_0000	Model Number (bits 0-2)
0x21E0_0000 - 0x21E0_03FE	PC/104 16-bit I/O (legacy support)
0x2100_0000	CF IDE 16-bit register
0x13C0_0000 - 0x13C0_0001	(TS-7300) FPGA loader registers
0x1340_0000 - 0x1340_0002	(TS-7260) Additional COM5 registers
0x1300_0000 - 0x1300_0002	(TS-7260) Additional COM4 registers
0x12C0_0000 - 0x12C0_0003	(TS-7260) DIO2 Header TS-XDIO registers
0x12C0_0000 - 0x12C0_0001	(TS-7260) DIO2 Header Basic DIO registers
0x1240_0000 - 0x1240_0002	(TS-7260) COM3 serial port registers
0x1200_0000	(TS-7260) Power Management register
0x11E0_0000 - 0x11E0_03FF	PC/104 8-bit I/O (legacy support)
0x11A0_0000 - 0x11AF_FFFF	PC/104 8-bit Memory (legacy support)
0x1170_0000	RTC R/W data register
0x1100_0001 - 0x1100_0007	CF IDE 8-bit registers
0x10F0_0000 - 0x10F0_0001	MAX197 A/D registers
0x1080_0000	MAX197 A/D busy bit (bit 7) (RO)
0x1080_0000	JP2-JP5 (bits 0,1,3,4) (RO)
0x1080_0000	COM1 DCD (bit 6) (RO)
0x1080_0000	Write Only index register
0x1040_0006 - 0x1040_0007	CF AUX IDE 8-bit registers

## 20.3 APENDICE C: Programas

### 20.3.1 Programa EasyBattery

```
# -*- coding: cp1252 -*-
```

```
#!/usr/bin/python
```

```
"""
```

```
=====
Name      : EasyBattery.py
Autor     : Maria Jesus Gutierrez
Versión   : 1.0
Descripción : EasyBattery in Python
```

```
=====
===== """
```

```
import wx
import time
import wx.gizmos as gizmos
import socket
import os, sys
import wx.aui
from wx.lib.wordwrap import wordwrap
from PIL import Image
```

```
#Inicializacion de variables globales
FONTSIZE = 12
C=90
FlagC=0
FlagR=0
FlagOff=0
client_socket=""
Vsc_g='---'
Vtc_g='---'
Ip_t='---'
Tdh_g='---'
percent_g='---'
ParaImprimir=""
Flag1=0
Flag_una_vez=0
```

```
class MyDialogOnCapacity(wx.Dialog):
```



```

def __init__(self, parent, id, title):

    wx.Dialog.__init__(self, parent, id, title, size=(460, 360),
                        pos=(430,150))

    #Formato de texto Socket Red
    fontCpercent = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                           u'Comic Sans')
    self.cpercent = wx.StaticText(self, -1, "", (40, 350), (60, -1))

    #inicialmente no hay datos recibidos, valor por defecto si no hay datos
    data_r='Sin datos'

    if (FlagC ==1):
        #socket
        data = '5'
        client_socket.send(data)
        print "ENVIADO:" ,data
        data = client_socket.recv (512)
        print "RECIBIDO:" ,data
        data_r = data

    else:

        print data_r

    #texto aviso si data no recibe ningun dato
    if (data_r <> 'Sin datos'):
        textAviso = wx.StaticText(self, -1, data_r, (140,325))
        Cpercent=data_r
        data_r="" #evita que se repita la información
        Cpercent= int(Cpercent)
        print 'Capacidad estimada %d' % (Cpercent)
        if(Cpercent==100):
            print 100
            percent=100
            vbleDraw='bateria completaR.png'
            fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                                u'Comic Sans')

            textAviso = wx.StaticText(self, -1,
                                     'Capacidad estimada 100%.\n Batería cargada',
                                     (110,240), style=wx.ALIGN_CENTRE)

            textAviso.SetFont(fontAviso)

        else:
            if(Cpercent==75):
                print 75
                vbleDraw='bateria a medias 1R.png'

```

```

fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                    u'Comic Sans')

textAviso = wx.StaticText(self, -1,
                          'Capacidad estimada 75%',
                          (125,240), style=wx.ALIGN_CENTRE)

textAviso.SetFont(fontAviso)

else:
    if(Cpercent==50):
        print 50
        vbleDraw='bateria a medias 4R.png'
        fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                            u'Comic Sans')

        textAviso = wx.StaticText(self, -1,
                                  'Capacidad estimada 50%',
                                  (125,240), style=wx.ALIGN_CENTRE)

        textAviso.SetFont(fontAviso)
    else:
        if(Cpercent==25):
            print 25
            vbleDraw='bateria a medias 7R.png'
            fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                                u'Comic Sans')

            textAviso = wx.StaticText(self, -1,
                                      'Capacidad estimada 25%',
                                      (125,240), style=wx.ALIGN_CENTRE)
            textAviso.SetFont(fontAviso)
        else:
            if(Cpercent==0):
                print 0
                vbleDraw='bateria a medias 10R.png'
                fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                                    u'Comic Sans')

                textAviso = wx.StaticText(self, -1,
                                          'Capacidad estimada 0%.\n Recargue la batería cuanto antes',
                                          (80,240), style=wx.ALIGN_CENTRE)
                textAviso.SetFont(fontAviso)

            else:
                print 'Error durante la estimación de la capacidad'

                textAviso = wx.StaticText(self, -1,
                                          'Error durante la estimación de la capacidad\n',
                                          (30,140), style=wx.ALIGN_CENTRE)
                textAviso.SetFont(fontAviso)
  
```

```

#imagenes
self.bitmap = wx.Bitmap(vbleDraw)
wx.EVT_PAINT(self, self.OnPaint)

else:
    bitmap = wx.StaticBitmap(self, -1, wx.Bitmap('escudos1.jpg'), pos=(100, 30))
    #en caso de que no este conectado con la tarjeta TS-7200
    #Formato del texto de aviso No hay conexión con la tarjeta TS-7200
    fontAviso = wx.Font(16, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                        u'Comic Sans')

    textAviso = wx.StaticText(self, -1,
                              'No hay conexión con la tarjeta TS-7200',
                              (30, 170), style=wx.ALIGN_CENTRE)

    textAviso.SetFont(fontAviso)

    #Formato del texto de aviso Compruebe batería vehiculo
    fontAviso1 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False,
                        u'Comic Sans')

    textAviso1 = wx.StaticText(self, -1,
                              "*Compruebe que la batería del vehículo este encendida...\n"
                              "*¿Ha pulsado establecer comunicación?\n", (30, 200))

    textAviso1.SetFont(fontAviso1)

    #parametros del botón Cerrar
    self.ButtonCerrar = wx.Button(self, id=17, label='Cerrar',
                                  pos=(145, 255), size=(175, 35)) #Título del botón

    self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OnCerrar)
    self.ButtonCerrar.SetToolTip(wx.ToolTip("Salir de esta pantalla")) #cartel de aviso

def OnCerrar(self, event):
    self.Close(True)

def OnPaint(self, event):
    dc = wx.PaintDC(self)
    dc.DrawBitmap(self.bitmap, 40, 10) #posición del escudo

#Función para presentar una tabla de datos
class TestListCtrlDialog(wx.Dialog):

    def __init__(self, parent, id, title):

        wx.Dialog.__init__(self, parent, id, title, size=(550, 250),

```

```

    pos=(430,150))

#Definir el color de fondo del panel
self.SetBackgroundColour(wx.Colour(128, 255, 0))#verde

# variables globales
global Vsc_g
global Vtc_g
global Ip_t
global Tdh_g
global ParaImprimir
global Flag1
global Flag_una_vez

if (FlagC ==1):
    if (Flag1==1):
        #Dia
        #actualizar con el tiempo del PC
        actualizarDate = time.localtime(time.time())
        #definir fecha
        stF =time.strftime("%d.%m.%y", actualizarDate)

        #Hora
        #actualizar con el tiempo del PC
        actualizar = time.localtime(time.time())
        #definir hora
        st = time.strftime("%H:%M:%S", actualizar)

        if(Flag_una_vez==1):
            ParaImprimir= " --, --,  0,  %s,  %s" % (st,stF)
            Flag_una_vez=0
        else:
            ParaImprimir = " SA,  SA,  SA,  %s,  %s" %(st,stF)

    self.SetBackgroundColour(wx.Colour(228, 228,228))

#parametros ventana
bitmap3 = wx.StaticBitmap(self,-1,wx.Bitmap('escudo
grande.jpg'),pos=(10,10),style=wx.SIMPLE_BORDER)
fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                    u'Comic Sans')

textAviso = wx.StaticText(self, -1,
                          'Capacidad estimada 0%.\n Recargue la batería cuanto antes',
                          (150,80), style=wx.ALIGN_CENTRE)
textAviso.SetFont(fontAviso)

#parametros del botón Cerrar
self.ButtonCerrar =wx.Button(self, id=17, label='Cerrar',
                             pos=(220, 155), size=(175,35))#Título del botón
  
```

```

self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OnCerrar)
self.ButtonCerrar.SetToolTip(wx.ToolTip("Salir de esta pantalla"))#cartel de aviso

else:

```

```

#Dia
#actualizar con el tiempo del PC
actualizarDate = time.localtime(time.time())
#definir fecha
stF =time.strftime("%d.%m.%y", actualizarDate)

#Hora
#actualizar con el tiempo del PC
actualizar = time.localtime(time.time())
#definir hora
st = time.strftime("%H:%M:%S", actualizar)

#Todas las variables globales necesarias se convierten a tipo
#string
S_Vsc_g=str(Vsc_g)
S_Vtc_g=str(Vtc_g)
S_Ip_t=str(Ip_t)
S_Tdh_g=str(Tdh_g)
S_percent_g=str(percent_g)
S_stF=str(stF)
S_st=str(st)

#se inicializa el indice a cero
self.index = 0

#Se crea una List Control
self.list_ctrl = wx.ListCtrl(self, style=wx.LC_REPORT
                             |wx.SUNKEN_BORDER|wx.LC_HRULES)

#Titulos de las columna
self.list_ctrl.InsertColumn(0, "Corriente nominal de consumo Cc (A)")
self.list_ctrl.InsertColumn(1, "Tiempo de descarga Td (h)")
self.list_ctrl.InsertColumn(2, "Porcentaje de carga (%)")
self.list_ctrl.InsertColumn(3, "Hora")
self.list_ctrl.InsertColumn(4, "Fecha")

#Items
for n in range(0,1):

    data_list = {

        n: (S_Ip_t,S_Tdh_g,S_percent_g, st,S_stF),
        #1 : ("", "", "", "", ""), #Si se desean añadir mas items

```

```

    }

    items = data_list.items()
    index = 0
    for key, i in items:
        self.list_ctrl.InsertStringItem(index, i[0])
        self.list_ctrl.SetStringItem(index, 1, i[1])
        self.list_ctrl.SetStringItem(index, 2, i[2])
        self.list_ctrl.SetStringItem(index, 3, i[3])
        self.list_ctrl.SetStringItem(index, 4, i[4])
        self.list_ctrl.SetItemData(index, key)
        index += 1

    self.itemDataMap = data_list

    #Imprimir datos
    ParaMostrar = str(" ".join([self.list_ctrl.GetItem(n, col).GetText() for col in
range(self.list_ctrl.GetColumnCount())]))

    if(Flag_una_vez==1):
        ParaImprimir=ParaMostrar
        Flag_una_vez=0
    else:
        ParaImprimir = " SA,  SA,  SA,  %s,  %s" %(st,stF)

    print ParaImprimir


    #parametros del botón Cerrar
    self.ButtonCerrar =wx.Button(self, id=17, label='Cerrar',
                                pos=(200, 150), size=(175,28))#Título del botón
    self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OffCerrar)
    #cartel de aviso
    self.ButtonCerrar.SetToolTip(wx.ToolTip("Salir de Datos"))

    #Formato
    sizer = wx.BoxSizer(wx.VERTICAL)
    sizer.Add(self.list_ctrl, 0, wx.ALL|wx.EXPAND, 10)
    self.SetSizer(sizer)

else:
    bitmap = wx.StaticBitmap(self,-1,wx.Bitmap('escudo grande.jpg'),pos=(5,5),)
    self.SetBackgroundColour(wx.Colour(228, 228,228))
    #en caso de que no este conectado con la tarjeta TS-7200
    #Formato del texto de aviso No hay conexión con la tarjeta TS-7200

```

```

fontAviso = wx.Font(16, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                    u'Comic Sans')

textAviso = wx.StaticText(self, -1,
                          'No hay conexión con la tarjeta TS-7200',
                          (130,50), style=wx.ALIGN_CENTRE)

textAviso.SetFont(fontAviso)

#Formato del texto de aviso Compruebe batería vehiculo
fontAviso1 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False,
                    u'Comic Sans')

textAviso1 = wx.StaticText(self, -1,
                          "*Compruebe que la batería del vehículo este encendida...\n"
                          "*¿Ha pulsado establecer comunicación?\n", (130, 100))

textAviso1.SetFont(fontAviso1)

#parametros del botón Cerrar
self.ButtonCerrar =wx.Button(self, id=17, label='Cerrar',
                             pos=(200, 155), size=(175,35))#Título del botón

self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OnCerrar)
self.ButtonCerrar.SetToolTip(wx.ToolTip("Salir de esta pantalla"))#cartel de aviso

#Dia
#actualizar con el tiempo del PC
actualizarDate = time.localtime(time.time())
#definir fecha
stF =time.strftime("%d.%m.%y", actualizarDate)

#Hora
#actualizar con el tiempo del PC
actualizar = time.localtime(time.time())
#definir hora
st = time.strftime("%H:%M:%S", actualizar)

if (Flag_una_vez==0):
    ParaImprimir= " SC,  SC,  SC,  %s,  %s" % (st,stF)

else:
    ParaImprimir = " SA,  SA,  SA,  %s,  %s" % (st,stF)

def OnCerrar(self, event):
    self.Close(True)

def OffCerrar(self, event):
    self.Close(True)

```



```
#Configuraciones de impresión, vista previa y configuraciones
class TextDocPrintout( wx.Printout ):

    def __init__( self, text, title, margins ):
        wx.Printout.__init__( self, title )
        self.lines = text.split( '\n' )
        self.margins = margins

    def HasPage( self, page ):
        return page <= self.numPages

    def GetPageInfo( self ):
        return ( 1, self.numPages, 1, self.numPages )

#Escala
def CalculateScale( self, dc ):
    ppiPrinterX, ppiPrinterY = self.GetPPIPrinter()
    ppiScreenX, ppiScreenY = self.GetPPIScreen()
    logScale = float( ppiPrinterX )/float( ppiScreenX )
    pw, ph = self.GetPageSizePixels()
    dw, dh = dc.GetSize()
    scale = logScale * float( dw )/float( pw )
    dc.SetUserScale( scale, scale )
    self.logUnitsMM = float( ppiPrinterX )/( logScale*25.4 )

#Configuración de la página
def CalculateLayout( self, dc ):
    topLeft, bottomRight = self.margins
    dw, dh = dc.GetSize()
    self.x1 = topLeft.x * self.logUnitsMM
    self.y1 = topLeft.y * self.logUnitsMM
    self.x2 = ( dc.DeviceToLogicalXRel( dw ) - bottomRight.x *
                self.logUnitsMM )
    self.y2 = ( dc.DeviceToLogicalYRel( dh ) - bottomRight.y *
                self.logUnitsMM )
    self.pageHeight = self.y2 - self.y1 - 2*self.logUnitsMM
    font = wx.Font( FONTSIZE, wx.TELETYPE, wx.NORMAL, wx.NORMAL )
    dc.SetFont( font )
    self.lineHeight = dc.GetCharHeight()
    self.linesPerPage = int( self.pageHeight/self.lineHeight )

def OnPreparePrinting( self ):
    dc = self.GetDC()
    self.CalculateScale( dc )
```

```

self.CalculateLayout( dc )
self.numPages = len(self.lines) / self.linesPerPage
if len(self.lines) % self.linesPerPage != 0:
    self.numPages += 1

```

```

def OnPrintPage(self, page):
    dc = self.GetDC()
    self.CalculateScale(dc)
    self.CalculateLayout(dc)
    dc.SetPen(wx.Pen("black", 0))
    dc.SetBrush(wx.TRANSPARENT_BRUSH)
    r = wx.RectPP((self.x1, self.y1), (self.x2, self.y2))
    dc.DrawRectangleRect(r)
    dc.SetClippingRect(r)
    line = (page-1) * self.linesPerPage
    x = self.x1 + self.logUnitsMM
    y = self.y1 + self.logUnitsMM
    while line < (page * self.linesPerPage):
        dc.DrawText(self.lines[line], x, y)
        y += self.lineHeight
        line += 1
    if line >= len(self.lines):
        break
    return True

```

#Establece comunicación con la tarjeta TS-7200

```
class MyDialogSetCommunication(wx.Dialog):
```

```
    def __init__(self, parent, id, title):
```

```
        wx.Dialog.__init__(self, parent, id, title, size=(300, 200))
```

```
        #variables globales
```

```
        global client_socket
```

```
        global FlagR
```

```
        global FlagC
```

```
        global FlagOff
```

```
        try:
```

```
            client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
            client_socket.connect(("192.168.0.50", 5000))
```

```
            data = client_socket.recv(512)
```

```
            print "RECIBIDO:" ,data
```

```
            print data
```

```
            if (data == 'Bienvenido a la placa TS-7200\n'):
```

```
                data=""
```

```
                FlagR=1
```

```
                FlagC=1
```

```
                FlagOff=1
```

```
                print "flags activados en la recepción"
```

```

diaSCd = wx.MessageDialog(None, 'Conectado con el servidor','Conectado a la
tarjeta TS-7200', wx.OK |wx.ICON_INFORMATION )
diaSCd.ShowModal()
diaSCd.Destroy()

if (data <> ""):
    FlagR=1
    FlagC=1
    #FlagOff=1
    print "Flags activos - Transmisión de datos"
    """if (data == 'Bienvenido a la placa TS-7200\n'):
        data= 'Conectado con Easybattery'
        client_socket.send(data)
        data = 'q'
        FlagR=0
        FlagC=0
        FlagOff==0
        Client.socket.close()
    else:"""

    if (data <> 'Q' and data <> 'q'):
        client_socket.send(data)
        FlagOff=1

    else:
        client_socket.send(data)
        data = 'q'
        FlagR=0
        FlagC=0
        FlagOff==0
        Ip_t="---"
        Tdh_g="---"
        Client.socket.close()

except:
    FlagR=0
    FlagC=0
    FlagOff=0
    print "No hay conexión con el servidor"

    #Mensaje de sin conexión
    dial = wx.MessageDialog(None, 'No es posible establecer conexión con el servidor',
        'Sin conexión', wx.OK | wx.ICON_ERROR)

    dial.ShowModal()

    #Mensaje de comprobaciones de sin conexión
    textAviso = 'No hay conexión con el servidor. Pruebe a realizar las siguientes
operaciones:\
    \n*Compruebe que el servidor esta encendido\
    \n*Compruebe que la batería del vehiculo este encendida'

```

```
diam = wx.MessageDialog(None, textAviso, 'Sin conexión',
                        wx.OK | wx.ICON_EXCLAMATION)
diam.ShowModal()
diam.Destroy()

#cerrar ventana
def OnCerrar(self, event):
    self.Close(True)

#Proporciona al interfaz de usuario
#informacion sobre si esta conectado a la red
class MyDialogRed(wx.Dialog):

    def __init__(self, parent, id, title):

        wx.Dialog.__init__(self, parent, id, title, size=(470, 410))

        #Variables globales
        global client_socket
        global FlagR

        #Formato de texto Socket Red
        fontRed = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                          u'Comic Sans')
        self.red = wx.StaticText(self, -1, "", (40, 75), (60, -1))
        #inicialmente no hay datos recibidos, valor por defecto si no hay datos:
        data_r = "Sin datos"

        if (FlagR == 1):
            #socket
            data = '2'
            client_socket.send(data)
            print "ENVIADO:" ,data
            data = client_socket.recv (512)
            print "RECIBIDO:" ,data
            data_r = data
            print 'data_r = %s' %data_r
        else:
            data_r = "Sin datos"

        #texto aviso si data no recibe ningun dato
        if (data_r <> "Sin datos"):
            if (data_r==" El sistema está conectado a la red\n"):
                fontAviso = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
                                    u'Comic Sans')
                textAviso1 = wx.StaticText(self, -1, data_r, (100,280))
```

```

    textAviso1.SetFont(fontAviso)
    print data_r

    bitmap1 = wx.StaticBitmap(self,-
1,wx.Bitmap('enchufadoR.png'),size=(400,230),pos=(30,30),style=wx.SIMPLE_BORDER)
    bitmap = wx.StaticBitmap(self,-1,wx.Bitmap('escudo
grande.jpg'),pos=(10,10),style=wx.SIMPLE_BORDER)
    else:
        fontAviso = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
            u'Comic Sans')
        textAviso2 = wx.StaticText(self, -1, data_r, (75,270), style=wx.ALIGN_CENTRE)
        textAviso2.SetFont(fontAviso)
        print data_r
        bitmap1 = wx.StaticBitmap(self,-
1,wx.Bitmap('desenchufadoR.jpg'),size=(280,250),pos=(150,10),style=wx.SIMPLE_BORDER)
        bitmap = wx.StaticBitmap(self,-1,wx.Bitmap('escudo
grande.jpg'),pos=(10,10),style=wx.SIMPLE_BORDER)

    else:
        #en caso de que no este conectado con la tarjeta TS-7200
        bitmap = wx.StaticBitmap(self,-1,wx.Bitmap('escudos1.jpg'),pos=(100,50))

        #Formato del texto de aviso No hay conexión con la tarjeta TS-7200
        fontAviso = wx.Font(16, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
            u'Comic Sans')

        textAviso = wx.StaticText(self, -1,
            'No hay conexión con la tarjeta TS-7200',
            (30,200), style=wx.ALIGN_CENTRE)

        textAviso.SetFont(fontAviso)

        #Formato del texto de aviso Compruebe batería vehiculo
        fontAviso1 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False,
            u'Comic Sans')

        textAviso1 = wx.StaticText(self, -1,
            "*Compruebe que la batería del vehículo este encendida...\n"
            "*¿Ha pulsado establecer comunicación?\n", (30,250))

        textAviso1.SetFont(fontAviso1)

        #parametros del botón Cerrar
        self.ButtonCerrar =wx.Button(self, id=17, label='Cerrar',
            pos=(145, 327), size=(175,35))#Título del botón

```

```
self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OnCerrar)
self.ButtonCerrar.SetToolTip(wx.ToolTip("Salir de esta pantalla"))#cartel de aviso
```

```
def OnCerrar(self, event):
    self.Close(True)
```

```
#Selecciona la capacidad de la batería actual
class MyDialogSelectBattery(wx.Dialog):
```

```
    def __init__(self, parent, id, title):

        wx.Dialog.__init__(self, parent, id, title, size=(300, 210))

        self.cfg = wx.Config('myconfig')

        wx.StaticText(self, -1,
                       'Seleccione la nueva capacidad de la batería adicional',
                       (15,20))

        wx.StaticText(self, -1, 'Capacidad de la batería (Ah):', (15,60))
        self.sc1 = wx.SpinCtrl(self, -1, str(C), (175, 56), (60, -1),
                               min=30, max=100)

        wx.Button(self, 1, 'Guardar', (100, 120))
        self.Bind(wx.EVT_BUTTON, self.OnSave, id=1)
```

```
def OnSave(self, event):
    #Variable global
    global C #Cambia el valor seleccionado en todo el programa

    self.cfg.WriteInt("Capacidad de la batería", self.sc1.GetValue())
    self.sc1.GetValue()
    C = self.sc1.GetValue()
    print "La capacidad de la batería actual es %s Ah" %C
```

```
#Proporciona al interfaz de usuario:
#información sobre si el sistema esta conectado a la red
class MyDialogCorriente(wx.Dialog):
```

```
    def __init__(self, parent, id, title):

        wx.Dialog.__init__(self, parent, id, title, size=(510, 450))

        #Variables globales
        global client_socket
        global FlagC
```

```
global Vsc_g
global Vtc_g
global Ip_t
global Tdh_g
global percent_g
```

```
global Flag1
global Flag_una_vez
```

```
Flag1='1'
```

```
#Formato de texto Socket Red
```

```
fontCorriente = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False,  
                        u'Comic Sans')
```

```
self.corriente = wx.StaticText(self, -1, "", (40, 75), (60, -1))
```

```
#inicialmente no hay datos recibidos, valor por defecto si no hay datos
```

```
data_r='Sin datos'
```

```
if (FlagC ==1):
```

```
    #socket corriente
```

```
    data = '6'
```

```
    client_socket.send(data)
```

```
    print "ENVIADO:" ,data
```

```
    data = client_socket.recv (512)
```

```
    print "RECIBIDO:" ,data
```

```
    data_r = data
```

```
    #socket capacidad estimada
```

```
    data = '5'
```

```
    client_socket.send(data)
```

```
    print "ENVIADO:" ,data
```

```
    data = client_socket.recv (512)
```

```
    print "RECIBIDO:" ,data
```

```
    data_r1 = data
```

```
else:
```

```
    data_r='Sin datos'
```

```
#texto aviso si data no recibe ningun dato
```

```
if (data_r <> 'Sin datos'):
```

```
    Ip=data_r
```

```
    Flag_una_vez=1
```

```
    data_r=""
```

```
    Ip= float(Ip)
```

```
#Ip representa el consumo electrico de los dispositivos
```

```
#TdhTiempo de descarga en horas-->Tiempo de duración de la batería
```

```
#en horas
```

```
#C Capacidad de la batería en Ah
```

```
#Tiempo de descarga
if(Ip==0):
    Tdh100=0
else:
    Tdh100=C/abs(Ip)
    print Tdh100

#percent actualizado
Cpercent=data_r1
data_r1="" #evita que se repita la información
Cpercent= int(Cpercent)
print ' Capacidad estimada %d' % (Cpercent)
if(Cpercent==100):
    print 100
    percent=100
    Flag1=0
else:
    if(Cpercent==75):
        print 75
        percent=75
        #self.OnPercent(percent)
        Flag1=0
    else:
        if(Cpercent==50):
            print 50
            percent=50
            #self.OnPercent(percent)
            Flag1=0

        else:
            if(Cpercent==25):
                print 25
                percent=25
                #self.OnPercent(percent)
                Flag1=0

            else:
                if(Cpercent==0):
                    print 0
                    percent=0
                    Flag1=1

            else:
                print 'No hay corriente en la batería'
                Flag1=1
```

```
#Tdh es el tiempo si la batería está al 100%
```



```
#adapta el valor al porcentaje de carga de la batería
```

```
percent_g=percent
```

```
Tdhx=(Tdh100*percent)/100
```

```
#los decimales limitados deben imponerse despues de utilizar
```

```
#el valor.
```

```
Tdhxd = '%.2f' % Tdhx
```

```
Tdh_g=str(Tdhxd)
```

```
#pasar a segundos
```

```
datos=Tdhx*3600
```

```
#-----RESULTADO-----
```

```
print "Batería de capacidad %s Ah" %C
```

```
print"Nota: Datos referidos a la capacidad nominal de la batería actual "
```

```
#Capacidad nominal de la batería
```

```
CTexto= "La batería adicional actual, tiene una capacidad de %s Ah." %C
```

```
fontAviso3 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,  
u'Comic Sans')
```

```
textAviso3 = wx.StaticText(self, -1, CTexto, (30,225))
```

```
textAviso3.SetFont(fontAviso3)
```

```
Ip_a= abs(Ip)
```

```
print Flag1
```

```
if (Flag1==1):
```

```
    Ip_t='--'
```

```
    bitmap2 = wx.StaticBitmap(self,-  
1,wx.Bitmap('bateriacomercialR.jpg'),size=(290,210),pos=(180,10),style=wx.SIMPLE_BORDER)
```

```
    bitmap3 = wx.StaticBitmap(self,-1,wx.Bitmap('escudo  
grande.jpg'),pos=(10,10),style=wx.SIMPLE_BORDER)
```

```
    fontAviso = wx.Font(14, wx.NORMAL, wx.NORMAL, wx.BOLD, False,  
u'Comic Sans')
```

```
    textAviso = wx.StaticText(self, -1,  
    'Capacidad estimada 0%.\n Recargue la batería cuanto antes',  
    (80,260), style=wx.ALIGN_CENTRE)
```

```
    textAviso.SetFont(fontAviso)
```

```
else:
```

```
    Ip_g = str(Ip_a)
```

```
    Ip_t=Ip_g
```

```
    print "Ip",Ip_g
```

```

    bitmap2 = wx.StaticBitmap(self,-
1,wx.Bitmap('bateriacomercialR.jpg'),size=(290,210),pos=(180,10),style=wx.SIMPLE_BORDER)

    bitmap3 = wx.StaticBitmap(self,-1,wx.Bitmap('escudo
grande.jpg'),pos=(10,10),style=wx.SIMPLE_BORDER)

    #Contenido del dato Ip
    IpTexto = "El consumo de corriente de las cargas es de: %s A." %Ip_t
    #Formato del texto del dato Ip
    fontAviso4 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
        u'Comic Sans')

    textAviso4 = wx.StaticText(self, -1, IpTexto, (30,250))
    textAviso4.SetFont(fontAviso4)

    #Impresión por pantalla
    print '%s %.2f %s' % ("Duración de la batería:", Tdhx, "horas")
    print "que equivalen a una duración aproximada de:"
    #Duración de la batería
    DuracionTexto = '%s %.2f %s' % ("Duración de la batería:", Tdhx,
        "horas.")

    #Formato del texto de Duración de la batería
    fontAviso5 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
        u'Comic Sans')

    textAviso5 = wx.StaticText(self, -1, DuracionTexto, (30,275))
    textAviso5.SetFont(fontAviso5)

    # Días.
    ndias, aux = divmod(datos, 86400)
    # Horas.
    nhoras, aux = divmod(aux, 3600)
    # Minutos y segundos.
    nmin, nseg = divmod(aux, 60)

    #Se muestra el resultado.
    print '%d dias , %d horas, %d minutos, %d segundos' %(ndias, nhoras, nmin, nseg)

    #self.logger.AppendText(" Click on object with Id %d\n" %event.GetId())
    tiempoDescargaTexto = "El tiempo de duración de la batería a este ritmo de
consumo\n es de:\t%d días, %d horas, %d minutos, %d segundos." %(ndias, nhoras, nmin, nseg)

    #Formato del texto del texto de Tiempo de descarga
    fontAviso2 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.BOLD, False,
        u'Comic Sans')
    textAviso2 = wx.StaticText(self, -1, tiempoDescargaTexto, (30,300))
    textAviso2.SetFont(fontAviso2)

```

```

else:
    # en caso de que no este conectado con la tarjeta TS-7200
    Flag_una_vez=0
    #Formato del texto del texto de Tiempo de descarga
    fontAviso3 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False, u'Comic
Sans')
    textAviso3 = wx.StaticText(self, -1, "Sin datos para calcular el tiempo de descarga",
(40,200))
    textAviso3.SetFont(fontAviso3)
    self.Center()

    #Formato del texto de aviso No hay conexión con la tarjeta TS-7200
    fontAviso = wx.Font(16, wx.NORMAL, wx.NORMAL, wx.BOLD, False, u'Comic
Sans')
    textAviso = wx.StaticText(self, -1, "No hay conexión con la tarjeta TS-7200\n\n"
, (40,230), style=wx.ALIGN_CENTRE)
    textAviso.SetFont(fontAviso)

    #Formato del texto de aviso Compruebe batería vehiculo
    fontAviso4 = wx.Font(12, wx.NORMAL, wx.NORMAL, wx.NORMAL, False, u'Comic
Sans')
    textAviso4 = wx.StaticText(self, -1, "POSIBLES SOLUCIONES:\n"
"*¿Ha pulsado establecer comunicación?\n"
"*Compruebe que la batería del vehiculo esta encendida...", (40,275))
    textAviso4.SetFont(fontAviso4)

    #escudo uc3m
    self.bitmap = wx.Bitmap('escudos1.jpg')
    wx.EVT_PAINT(self, self.OnPaint)
    self.Centre()

    #parametros del botón Cerrar Ventana
    self.ButtonCerrar =wx.Button(self, id=18, label='Cerrar',
pos=(160, 350), size=(175,35))#Título del botón

    self.ButtonCerrar.Bind(wx.EVT_BUTTON, self.OnCerrar)
    self.ButtonCerrar.SetToolTip(wx.ToolTip
("Salir de esta pantalla"))#cartel de aviso

    #Cerrar ventana
    def OnCerrar(self, event):
        self.Close(True)
    #Insertar bitmap
    def OnPaint(self, event):
        dc = wx.PaintDC(self)
        dc.DrawBitmap(self.bitmap, 120, 30) #posicion del escudo
  
```

```

#Menu principal
class MyMenu(wx.Frame):
    def __init__(self, parent, id, title):

        wx.Frame.__init__(self, parent, id, title, wx.DefaultPosition,
                           wx.Size(750, 650)) #tamano ventana

        #color de fondo
        self.SetBackgroundColour(wx.Colour(128, 255, 0))#verde

        #Menu principal
        menubar = wx.MenuBar()
        file = wx.Menu()
        datas = wx.Menu()
        help = wx.Menu()
        tiempo = wx.Menu()
        options = wx.Menu()

        #Desplegables del menu principal
        file.Append(101, '&Abrir\tCtrl+O', 'Abrir un documento')
        file.Append(102, '&Guardar\tCtrl+S', 'Guardar un documento')
        file.Append(107, '&Configurar Pagina...', 'Ajustes de impresion')
        file.Append(109, '&Vista previa', 'Vista previa de impresion ')
        file.Append(108, '&Imprimir\tCtrl+P', 'Imprimir pantalla')
        file.AppendSeparator()
        tiempo.Append(103, '&Fecha', 'Fecha')
        quit = wx.MenuItem(file, 105, '&Salir\tCtrl+Q', 'Salir de la aplicacion')
        quit.SetBitmap(wx.Image('stock_exit-16.png',
                                wx.BITMAP_TYPE_PNG).ConvertToBitmap())
        file.AppendItem(quit)
        options.Append(106, '&Fondo\tCtrl+F', 'Elegir Color de fondo')
        options.Append(111, '&Batería adicional\tCtrl+B', 'Seleccionar batería')
        help.Append(110, '&Acerca de EasyBattery', 'Información sobre el programa')
        datas.Append(112, '&Datos\tCtrl+D', 'Salida de datos')
        datas.Append(113, '&Eliminar', 'Elimina todos los datos pendientes de imprimir')
        datas.Append(114, '&Capacidad estimada\tCtrl+C', 'Muestra la capacidad de la batería en
tiempo real')

        #reloj de Leds
        pos = (400, 70)
        size = (300, 50)
        style = gizmos.LED_ALIGN_CENTER #usuario windows
        self.led = gizmos.LEDNumberCtrl(self, -1, pos, size, style)
        #por defecto son negro y verde

        #Hora
        #para usuarios windows en linux es diferente
        wx.CallAfter(self.OnTimer, None)
        self.timer = wx.Timer(self, -1)
        self.timer.Start(1000) #actualiza el reloj cada 1000ms
        self.Bind(wx.EVT_TIMER, self.OnTimer)

```

## #Botones

#botón para establecer la comunicación

```
wx.Button(self, 32, 'Establecer comunicación', (50,200), (150,40))
```

```
self.Bind(wx.EVT_BUTTON, self.OnSetComunication, id=32)
```

#botón para Cerrar la comunicación

```
wx.Button(self, 33, 'Cerrar comunicación', (50,250),(150,40))
```

```
self.Bind(wx.EVT_BUTTON, self.OnOffComunication, id=33)
```

#botón para comprobar el estado de la red

```
wx.Button(self, 30, 'Comprobar conexión a red', (50,400),(200,40))
```

```
self.Bind(wx.EVT_BUTTON, self.OnShowRed, id=30)
```

#botón para comprobar la intensidad demandada

```
wx.Button(self, 31, 'Comprobar intensidad demandada',(50,450),(200,40))
```

```
self.Bind(wx.EVT_BUTTON, self.OnShowCorriente, id=31)
```

## #Eventos menu

#evento menu para mostrar datos

```
self.Bind(wx.EVT_MENU, self.OnShowDatos, id= 112)
```

## #Textos estaticos

### #hora

```
font1 = wx.Font(20, wx.NORMAL, wx.NORMAL, wx.BOLD)
```

```
lyrics1 = wx.StaticText(self, -1, 'HORA', (490,30), style=wx.ALIGN_CENTRE)
```

```
lyrics1.SetFont(font1)
```

```
self.Center()
```

### #Menubar

```
menubar.Append(file, '&Archivo')
```

```
menubar.Append(datos, '&Datos')
```

```
menubar.Append(tiempo, '&Fecha')
```

```
menubar.Append(options, '&Opciones')
```

```
menubar.Append(help, '&Ayuda')
```

### #Eventos

```
wx.EVT_MENU(self, 101, self.OnOpen)
```

```
wx.EVT_MENU(self, 102, self.CreateFile)
```

```
wx.EVT_MENU(self, 103, self.OnDate)
```

```
wx.EVT_MENU(self, 105, self.OnClose)
```

```
wx.EVT_MENU(self, 106, self.choosecolor)
```

```
wx.EVT_MENU(self, 107, self.pagesetup)
```

```
wx.EVT_MENU(self, 108, self.OnPrinter)
```

```
wx.EVT_MENU(self, 109, self.OnPrintPreview)
```

```
wx.EVT_MENU(self, 110, self.OnAcercaDe)
```

```

wx.EVT_MENU(self, 111, self.OnShowBattery)
wx.EVT_MENU(self, 112, self.OnShowDatos)
wx.EVT_MENU(self, 113, self.OnClear)
wx.EVT_MENU(self, 114, self.OnCapacity)

self.SetMenuBar(menubar)
self.CreateStatusBar()

#escudo uc3m
self.bitmap = wx.Bitmap('escudos1.jpg')
wx.EVT_PAINT(self, self.OnPaint)
self.Centre()

#imprimir
self.pdata = wx.PrintData()
self.pdata.SetPaperId(wx.PAPER_LETTER)
self.pdata.SetOrientation(wx.PORTRAIT)
self.margins = (wx.Point(15,15), wx.Point(15,15))

#TextCtrl y Notebook
self.nb = wx.aui.AuiNotebook(self,-1, (300, 180), (400, 360),
                             style=wx.TE_MULTILINE|wx.TE_READONLY)

self.tc = wx.TextCtrl(self.nb, -1, "", (300, 300), (400, 200),
                      style=wx.TE_MULTILINE|wx.TE_READONLY|wx.HSCROLL)

self.nb.AddPage(self.tc, "Impresión de Datos ")
self.tc.SetFont(wx.Font(FONTSIZE, wx.TELETYPE, wx.NORMAL, wx.NORMAL))

#FUNCIONES
def OnCapacity(self, event):
    diaOC = MyDialogOnCapacity(self, -1, 'CAPACIDAD DE LA BATERÍA')
    diaOC.ShowModal()
    diaOC.Destroy()

def OnClear(self, event):
    #Borrar TextCtrl
    self.tc.Clear()

def OnShowDatos(self, event):
    #Muestra detos de la opcion Datos
    #Datos
    diaSD = TestListCtrlDialog(self, -1, 'VISUALIZACIÓN DE DATOS...')
    #Imprimir ParaImprimir
    self.tc.AppendText("\n")
    self.tc.AppendText(" Cc(A) Td(h) Pc(%) Hora Fecha\n ")
    self.tc.AppendText("-----\n ")
    self.tc.AppendText(ParaImprimir)
    self.tc.AppendText("\n")

```

```
diaSD.ShowModal()
diaSD.Destroy()
```

```
def OnItemSelected(self, event):
    #Seleccionar Item
    selected_row = event.GetIndex()
    val = list()
    for column in range(3):
        item = self.lst.GetItem(selected_row, column)
        val.append(item.GetText())
    # Muestra que fue seleccionado el item de Datos
    frame = self.GetTopLevelParent()
    frame.PushStatusText(".".join(val))
```

```
def OnClose(self, event):
    #Cerrar ventana principal
    self.Close()
```

```
def OnTimer(self, event):
    #actualizar con el tiempo del PC
    actualizar = time.localtime(time.time())
    #definir hora
    st = time.strftime("%H %M %S", actualizar)
    self.led.SetValue(st)
```

```
def OnDate (self, event):
    #Fecha
    #actualizar con el tiempo del PC
    actualizarDate = time.localtime(time.time())
    #definir fecha
    stF =time.strftime("%d.%m.%y", actualizarDate)
    #Fuente del texto
    font1 = wx.Font(25, wx.NORMAL, wx.NORMAL, wx.BOLD)
    lyrics1 = wx.StaticText(self, -1, stF, (480,130), style=wx.ALIGN_CENTRE)
    lyrics1.SetFont(font1)
    #Barra StatusText
    self.SetStatusText(stF)
```

```
def OnSetComunication(self, event):
    #Establecer comunicación
    diaSC = MyDialogSetComunication(self, -1, 'ESTABLECIENDO CONEXION...')
```

```
def OnOffComunication(self, event):
    #Cerrar comunicación
```

```
#Variables globales
global client_socket
global FlagOff
global FlagR
global FlagC
global client_socket
global Ip_g
global Tdh_g

if (FlagOff==0):
    dial = wx.MessageDialog(None, 'No puede cerrar una conexión que no se ha
iniciado.\nInicie conexión.', 'Conexión Fallida', wx.OK | wx.ICON_ERROR)
    dial.ShowModal()
else:
    data = 'q'
    client_socket.send(data)
    Ip_g="---"
    Tdh_g="---"
    FlagR=0
    FlagC=0
    FlagOff=0
    client_socket.close()
    print 'La conexión con la tarjeta TS-7200 ha sido cerrada'
    diaOC = wx.MessageDialog(None, 'La conexión con la tarjeta TS-7200 ha sido cerrada. ',
'CERRANDO CONEXION...', wx.OK | wx.ICON_EXCLAMATION )
    diaOC.ShowModal()
    diaOC.Destroy()

def OnShowRed(self, event):
    #Muestra informacion sobre si esta conectado a la red o no
    dia = MyDialogRed(self, -1, 'COMPROBANDO CONEXION A LA RED...')
    dia.ShowModal()
    dia.Destroy()

def OnShowCorriente(self, event):
    #Muestra informacion sobre la corriente consumida por la batería
    diaC = MyDialogCorriente(self, -1, 'COMPROBANDO INTENSIDAD DEMANDADA..')
    diaC.ShowModal()
    diaC.Destroy()

def OnPaint(self, event):
    #Escudos
    dc = wx.PaintDC(self)
    dc.DrawBitmap(self.bitmap, 10, 10) #posicion del escudo UC3m LSI

def choosecolor(self, event):
    #Selección del fondo de pantalla
    dlg = wx.ColourDialog(self)
    dlg.GetColourData().SetChooseFull(True)
```



```
if dlg.ShowModal() == wx.ID_OK: #si presiona Aceptar
    data = dlg.GetColourData()
    self.SetStatusText('Has seleccionado: %s\n' % str(data.GetColour().Get()))
    self.SetBackgroundColour(data.GetColour())
    #Actualizar color
    self.Refresh()
    dlg.Destroy()

def pagesetup(self, event):
    #configurar página
    data = wx.PageSetupDialogData()

    data.SetMarginTopLeft( (12, 12) )
    data.SetMarginBottomRight( (12, 12) )
    data.SetPaperId(wx.PAPER_A4)

    dlg = wx.PageSetupDialog(self, data)

    if dlg.ShowModal() == wx.ID_OK:
        data = dlg.GetPageSetupData()
        self.pdata = wx.PrintData(data.GetPrintData())
        self.pdata.SetPaperId(data.GetPaperId())
        self.margins = (data.GetMarginTopLeft(), data.GetMarginBottomRight())

    dlg.Destroy()

def OnPrinter(self, event):
    #imprimir

    dato = wx.PrintDialogData(self.pdata)
    dlg = wx.PrintDialog(self, dato)
    data = dlg.GetPrintDialogData()

    if dlg.ShowModal() == wx.ID_OK:

        printer = wx.Printer(data)
        #print "%s", printer
        text = self.tc.GetValue()
        printout = TextDocPrintout(text, "title", self.margins)
        useSetupDialog = False
        if not printer.Print(self, printout, useSetupDialog) \
            and printer.GetLastError() == wx.PRINTER_ERROR:
            wx.MessageBox(
                "Existe un problema de impresion.\n"
                "Tal vez la impresora actual no está bien conectada",
                "Error de impresion", wx.OK)
        else:
            data = printer.GetPrintDialogData()
            self.pdata = wx.PrintData(data.GetPrintData())
```

```
printout.Destroy()

def GetErrorText():
    #Muestra error interno
    print "Ocurrio error"

def OnPrintPreview(self, evt):
    #Vista previa
    data = wx.PrintDialogData(self.pdata)
    text = self.tc.GetValue()
    printout1 = TextDocPrintout(text, "title", self.margins)
    printout2 = TextDocPrintout(text, "title", self.margins)
    preview = wx.PrintPreview(printout1, printout2, data)
    if not preview.Ok():
        wx.MessageBox(" Deshabilitar para crear una vista preliminar!", "Error")
    else:
        frame = wx.PreviewFrame(preview, self, "Vista previa", pos=self.GetPosition(),
size=self.GetSize())
        frame.Initialize()
        frame.Show()

def CreateFile(self, event):
    #Crea un archivo

    #tipo de cuadro de dialogo y cabecera
    dlg = wx.FileDialog(self, "Guardar como: ", os.getcwd(), "", " *.*", wx.SAVE)#botón de
guardar

    if dlg.ShowModal() == wx.ID_OK:
        #Path
        path =dlg.GetPath()
        mypath = os.path.basename(path)
        self.SetStatusText("Guardado en : %s " %path)#aparece en el Statusbar

        #Contenido del archivo
        itcontains = self.tc.GetValue()

        #Nombre de archivo a guardar y directorio donde se encuentra
        self.filename = dlg.GetFilename()
        self.dirname= dlg.GetDirectory()
        #Creacion del archivo
        filehandle=open(os.path.join(self.dirname, self.filename),'w')
        #llenado del archivo
        filehandle.write(itcontains)
        #Cierre del archivo
        filehandle.close()

    dlg.Destroy()
```

```
def OnOpen(self, event):
    #Abre un archivo

    self.dirname = "
    #tipo de cuadro de dialogo y cabecera
    dlg = wx.FileDialog(self, "Guardar como: ", os.getcwd(), "", " *.*", wx.OPEN)#botón de
abrir

    if dlg.ShowModal() == wx.ID_OK:

        #Nombre de archivo a abrir y directorio donde se encuentra
        self.filename=dlg.GetFilename()
        self.dirname=dlg.GetDirectory()

        #Abrir el archivo
        filehandle=open(os.path.join(self.dirname, self.filename), 'r')
        self.tc.SetValue(filehandle.read())

        filehandle.read()

        #Cierre del archivo
        filehandle.close()

        #Informa del nombre del ultimo archivo leído
        self.SetStatusText("Editando ... "+self.filename)

    dlg.Destroy()

def OnAcercaDe(self,event):
    #Informacion sobre el programa
    info = wx.AboutDialogInfo()
    info.Name = "EasyBattery"
    info.Version = "versión 1.0"
    info.Copyright = "(C) 2011"
    info.Description = wordwrap(
        "Este programa ha sido desarrollado\npara"
        " la Universidad Carlos III en wxPython"
        " por: María Jesús Gutiérrez.\n"
        "e-mail: 100032451@uc3m.es\n"
        "\n",
        350, wx.ClientDC(self))
    info.WebSite =
("http://www.uc3m.es/portal/page/portal/dpto_ing_sistemas_automatica/investigacion/lab_sist_i
nteligentes")
    info.Developers = [ "Maria Jesus Gutierrez 2011" ]
    #La licencia se añadiría aquí
    #info.License = wordwrap(licenseText, 500, wx.ClientDC(self))
    #licenseText = ""

    wx.AboutBox(info)
```

```

def OnShowBattery(self, event):
    #Selección de la capacidad nominal
    diaB = MyDialogSelectBattery(self, -1, 'SELECCION CAPACIDAD')
    diaB.ShowModal()
    diaB.Destroy()

class MyApp(wx.App):
    def OnInit(self):
        frame = MyMenu(None, -1, 'EasyBattery.py')
        frame.SetIcon(wx.Icon('escudo pequeño.ico', wx.BITMAP_TYPE_ICO))
        frame.Show(True)
        self.SetTopWindow(frame)
        return True

app = MyApp(0)
app.MainLoop()

```

### 20.3.2 Programa TS-7200

```

/*
=====
Nombre      : Easybattery.c
Autor       : María Jesús Gutiérrez
Version     : 1.0
Descripción : Hello World in C, Ansi-style
=====
*/

#include <unistd.h>
#include <sys/types.h>
#include <sys/mman.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <assert.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <errno.h>

#include "peekpoke.h"
#include "ep93xx_adc.h"

#define DATA_PAGE    0x12C00000
#define CALIB_LOC     2027           //localización de los valores de
calibración
#define NUM_SAMPLES   10           //Numero de muestras
#define NUM_CHANNELS   2           //Número de canales

```

```
/* Declaración de variables globales */
static unsigned long adc_page, syscon_page;
char *dr_page;

char send_data[1024];
char recv_data [1024];
char dato_DIO_0 [1024];
double voltADC0;
double voltADC4;

/*Calculo correspondiente al valor del ADC a 0V*/
//val1 es el valor del adc correspondiente a 0.833V
//val2 es el valor del adc correspondiente a 2.5V
//Estos valores se obtienen de la curva de calibración. Fuente: Hojas
características del EP9302.
int calcZeroVal (int val1, int val2)
{
    val2 += 0x10000; /* En hexadecimal es: 2^16 (16 es el nº de bits) ver
curva de calibración.*/
    return (int)(val1-(((val2-val1)/(2.5-0.833))*0.833));
}
/* Los valores 2.5 y 0.833 se obtienen de la curva de calibración.*/
//La devolución de un valor 1 indica que la placa no tiene valores de
calibración.
//La devolución de un valor 0 indica que la placa tiene valores de
calibración.

void setDR(char *x,int n,int val)
{
    if (n < 0 || n > 8)
        return;

    x[0] = (x[0] & ~(1 << n)) | (val ? (1<<n) : 0);
}

void setD(char *x,int n,int val)
{
    if (n < 0 || n > 8)
        return;

    x[2] = (x[2] & ~(1 << n)) | (val ? (1<<n) : 0);
}

double get_volts(int val, int zero, int range)
{
    /*double voltADC0;
double voltADC4;
double fact_correct;

int i,j,avgt,avgm;

    for (i=0; i<2;i++)
    {
        avgt=0;
        for(j=0; j<NUM_SAMPLES; j++)
        {
            avgt=avg+avgt;
            avgm=avgt/10;
```

```

        if(avgm <= 0x7000)
        {
            avgm = avgm + 0x10000;
        }

        avgm = avgm - zero;
        fact_correct = 0.03; //Valor de ajuste de la tensión

        if(i = 0)
        {
            voltADC0 = avgm;
        }

        if (i=1)
        {
            voltADC4=avgm;
        }
    }
}

return (((double)voltADC0 * 3.3) / range)- fact_correct;
return (((double)voltADC4 * 3.3) / range)- fact_correct;
return (voltADC0);
return (voltADC4);

*/

double volt;
double fact_correct;
    if (val <= 0x7000)
        val = val + 0x10000;

    val = val - zero;
    fact_correct = 0.015; //Valor de ajuste de la tensión

    return (((double)val * 3.3) / range)- fact_correct;
    return (volt );
}

void calc_calibration(int calibration[0][2], int
adc_result_1[NUM_CHANNELS][NUM_SAMPLES])
{
    int i,j;
    i=0;
    for(i=0; i<NUM_CHANNELS; i++)
    {
        /* el cero esta fuera de los valores de calibración */

        for(j = 0; j < 2; j++)
        {

            calibration[i][j] = 0;

        }
    }
    for(i=0; i<NUM_CHANNELS; i++)
    {
        // convierte a 0v los valores convert 0.833V vals to 0V vals

```

```

        for(j=0; j<NUM_SAMPLES; j++)
        {
            adc_result_1[i][j] = calcZeroVal(adc_result_1[i][j],
adc_result_1[0][j]);

        }
    }

    //suma las lecturas
    for(i=0; i<NUM_CHANNELS; i++)
    {

        for(j = 0; j < NUM_SAMPLES; j++ )
        {
            calibration[i][0] = adc_result_1[i][j] +
calibration[i][0];

        }
    }
}

/*****
 *Lectura de la EP9302. Se descartan las dos primeras muestras y
 * se guarda el resto.
 *****/
static void read_7xxx_adc(int adc_result[NUM_CHANNELS][NUM_SAMPLES])
{
    int i, j, cur_ch;
    /*Aunque la EP9302 dispone de cinco canales la TS-7200 solo tiene dos
accesibles*/

    for(i = 0; i < NUM_CHANNELS; i++)
    {
        switch(i)
        {
            case 0:
                cur_ch = ADC_CH0;
                break;

            case 1:
                cur_ch = ADC_CH4;
                break;

        }

        //Descarte de las dos primeras muestras
        read_channel(adc_page, cur_ch);
        read_channel(adc_page, cur_ch);

        //Lectura del resto de muestras
        for(j = 0; j < NUM_SAMPLES; j++)
        {
            usleep(10000);
            adc_result[i][j] = read_channel(adc_page, cur_ch);
        }
    }
}

```

```
void print_adc_channel(int calibration[NUM_CHANNELS][2])
{
    int adc_result_1[NUM_CHANNELS][NUM_SAMPLES];

    setD(dr_page, 0, 1);    //ADC1 = ADC3 = 0.833V
    setD(dr_page, 2, 1);    //ADC0 = ADC2 = 2.5V

    read_7xxx_adc(adc_result_1);

    voltADC0 = get_volts(adc_result_1[0][0], 0x9E58, 0xC350);
    voltADC4 = get_volts(adc_result_1[1][0], 0x9E58, 0xC350);

    printf("El voltaje en del ADC_0 es: %3.2fV\n", voltADC0);
    printf("El voltaje en del ADC_4 es: %3.2fV\n", voltADC4);

}
void DI_func()
{
    volatile unsigned int *PBDR, *PBDDR, *GPIOBDB;

    unsigned char *start;

    unsigned char state;

    int fd = open("/dev/mem", O_RDWR|O_SYNC);
    start = mmap(0, getpagesize(), PROT_READ|PROT_WRITE, MAP_SHARED,
                 fd, 0x80840000);

    PBDR = (unsigned int *)(start + 0x04); //data register, del puerto b
    //puerto b, pag 532 EP9301 manual de usuario, 0x8084_0004
    PBDDR = (unsigned int *)(start + 0x14); //dirección del puerto b,
    //pag 536 EP9301 manual de usuario,
    0x8084_0014
    GPIOBDB = (unsigned int *)(start + 0xC4); // antirrebote pág 547 EP9301
    manual de usuario

    *PBDDR = 0xf0; //Todos los pines DIO salida menos el DIO_1 que es una entrada
    (1111_1101)
    // El DDR controla si cada pin DIO es una entrada o una salida
    //"1" es salida, "0" es entrada
    //Ejemplo
    //1111_0000 habilita como entradas DIO_0,DIO_1,DIO_2,DIO_3 y como salidas
    DIO_4, DIO_5, DIO_6, DIO_7

    *GPIOBDB = 0x01; //habilitar el rebote en el bit 0
    state = *PBDR; // lectura del estado inicial

    if (state & 0x01) // si el DIO_0 (bit 0 del puerto B) está a "1"
    {

        //printf ("state: %d\n", state);
        //printf(" DIO_0 es: uno :\n El ssistema está conectado \n");
    }
}
```



```

        sprintf(dato_DIO_0, " El sistema está conectado a la red\n ");
    }
    else
    {
        //printf ("state: %d\n",state);
        //printf(" DIO_0 es cero: el sistema no está conectado
a la red\n");
        sprintf(dato_DIO_0,"El sistema no está conectado a la
red\n");
    }

close(fd);
return;
}
void DO_func(int state_DO_DIO_7)
{
    volatile unsigned int *PBDR, *PBDDR;
    unsigned char *start;
    unsigned char read_o;

    int fd = open("/dev/mem", O_RDWR|O_SYNC);
    start = mmap(0, getpagesize(), PROT_READ|PROT_WRITE, MAP_SHARED,
                fd, 0x80840000);

    PBDR = (unsigned int *)(start + 0x04); //Registro de datos, del puerto b
    //puerto b, pág 532 EP9301 manual de usuario, 0x8084_0004
    PBDDR = (unsigned int *)(start + 0x14); //dirección del puerto b,
    //pag 536 EP9301 manual de
usuario, 0x8084_0014

    *PBDDR = 0xf0;
    /*Todos los pines DIO salida menos el DIO_1 que es una entrada
(1111_1101)
    El DDR controla si cada pin DIO es una entrada o una salida
    "1" es salida, "0" es entrada
    Ejemplo
    1111_0000 habilita como entradas DIO_0,DIO_1,DIO_2,DIO_3 y como salidas
DIO_4, DIO_5, DIO_6, DIO_7
    */

    if (state_DO_DIO_7 == 2)
    {
        *PBDR = *PBDR | 0x80; //fuerzo el bit a 1
        printf ("salida 4000 ");
    }

    if (state_DO_DIO_7 == 3)
    {
        *PBDR = *PBDR & 0x7f; //fuerzo el bit a 0
        printf ("salida 00 ");
    }

    read_o=*PBDR;

    if (read_o & 0x80)
        printf ("el bit 8 es 1\n");
    else

```

```
        printf ("el bit 8 es 0\n");

        close(fd);
        return;
    }

    int main(void)
    {
        int calibration[NUM_CHANNELS][2];

        int ret_val;
        int code;

        float Ip;
        float Vtc;
        int Cpercent;

        int devmem = open("/dev/mem", O_RDWR|O_SYNC);
        assert(devmem != -1);

        int sock, connected, bytes_recieved , true = 1;

        struct sockaddr_in server_addr, client_addr;
        int sin_size;

        void print_adc_channel(int calibration[NUM_CHANNELS][2]);
        void DI_func();
        void DO_func();

        dr_page = mmap(0, getpagesize(), PROT_READ|PROT_WRITE,
            MAP_SHARED, devmem, DATA_PAGE);
        assert(&dr_page != MAP_FAILED);
        adc_page = (unsigned long)mmap(0, getpagesize(), PROT_READ|PROT_WRITE,
            MAP_SHARED, devmem, ADC_PAGE);
        assert(&adc_page != MAP_FAILED);
        syscon_page = (unsigned long)mmap(0, getpagesize(), PROT_READ|PROT_WRITE
            , MAP_SHARED, devmem, SYSCON_PAGE);
        assert(&syscon_page != MAP_FAILED);

        init_ADC(adc_page, syscon_page);
        setDR(dr_page, 0, 1);
        setDR(dr_page, 1, 0);
        setDR(dr_page, 2, 1);
        setDR(dr_page, 3, 0);

        printf("Bienvenido a la placa TS-7200\n");

        //Se crea el socket con protocolo TCP/IP
        //sock es el identificador de socket
        if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
        {
            perror("Error socket");
        }
    }
}
```

```
        exit(1);
    }

    if (setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &true, sizeof(int)) == -1)
    {
        perror("Error setsockopt");
        exit(1);
    }
    //Asignación de una variable asignar valores a una variable de tipo
    sockaddr_in
    server_addr.sin_family = AF_INET;    //protocolo TCP/IP
    server_addr.sin_port = htons(5000);    //los paquetes son recibidos por el
    puerto 5000
    server_addr.sin_addr.s_addr = INADDR_ANY; //Dirección IP donde se reciben los
    paquetes se asigna de forma automática

    bzero(&(server_addr.sin_zero), 8);

    if (bind(sock, (struct sockaddr *)&server_addr, sizeof(struct sockaddr))
        == -1)
    {
        perror("Error en bind");
        exit(1);
    }
    //Habilitación del socket para poder recibir conexiones
    //Permite un máximo de 5 conexiones en cola de entrada de conexiones
    if (listen(sock, 5) == -1)
    {
        perror("Error listen");
        exit(1);
    }

    printf("\nServidor TCP esperando a cliente por puerto 5000");
    fflush(stdout);

    /*while(1)
    {
        /*
        //Inicio de la conexión
        sin_size = sizeof(struct sockaddr_in);
        connected = accept(sock, (struct sockaddr *)&client_addr, &sin_size);

        //Informa desde el puerto y dirección IP desde donde se consiguió la
        conexión.
        printf("\n La conexión se consiguió desde (%s , %d)",
            inet_ntoa(client_addr.sin_addr), ntohs(client_addr.sin_port));

        //Envía al cliente un mensaje de bienvenida
        sprintf(send_data, "Bienvenido a la placa TS-7200\n");
        send(connected, send_data, strlen(send_data), 0);

        //Muestra la información enviada
        printf("\n DATO ENVIADO = %s " , send_data);

        while (1)
        {
            //Recepción de datos
            bytes_recieved = recv(connected, recv_data, 1024, 0);
            recv_data[bytes_recieved] = '\0';
            if (strcmp(recv_data , "q") == 0 || strcmp(recv_data , "Q") == 0)
```

```

    {
        close(connection);
        break;
    }

    else
        printf("\n DATO RECIBIDO = %s " , recv_data);
        code =atoi (recv_data);

        /*if (strcmp(recv_data , "0") == 0 || strcmp(recv_data ,
"1") == 0)

            {

                if (strcmp(recv_data , "0") == 0)
                {
                    code=0;
                }
                if (strcmp(recv_data , "1") == 0)
                {
                    code=1;
                }

            }

            else

                close(connection);

        */

switch (code)
{
    case 0 :
        printf("Seleccionado codigo 0\n");
        printf("Bienvenido a la tarjeta TS-7200\n");
        sprintf(send_data, "Bienvenido a la tarjeta TS-7200");
        send(connection, send_data,strlen(send_data), 0);
        fflush(stdout);
        fflush(stdout);

        break;

    case 1 :
        printf("Seleccionado codigo 1\n");
        print_adc_channel( calibration[NUM_CHANNELS][2]);
        //sprintf(send_data, "El voltaje en del ADC_0 es:
%3.2fV\n", voltADC0);

        sprintf(send_data, "%3.2f", voltADC0);
        send(connection, send_data,strlen(send_data), 0);

        fflush(stdout);
        break;

    case 2 :
        printf("Seleccionado codigo 2\n");
        DI_func();

```

```

sprintf(send_data, dato_DIO_0);
printf("\n DATO ENVIADO = %s " , send_data);
send(connection, send_data, strlen(send_data), 0);
fflush(stdout);
    break;

case 3:

printf("Seleccionado codigo 3\n");
DO_func(2);
sprintf(send_data, " La salida se ha activado");
send(connection, send_data, strlen(send_data), 0);
fflush(stdout);

    break;

case 4:

    printf("Seleccionado codigo 4\n");
    DO_func(3);
    sprintf(send_data, " La salida se ha
desactivado");

    send(connection, send_data, strlen(send_data), 0);
    fflush(stdout);

    break;

case 5 :

printf("Seleccionado codigo 5\n");
print_adc_channel(
calibration[NUMBER_CHANNELS][2]);
if(voltADC4>=2.94)
    Cpercent=100;
else
    if(voltADC4>=2.88)
        Cpercent=75;
    else
        if(voltADC4>=2.81)
            Cpercent=50;

        else
            if(voltADC4>=2.73)
                Cpercent=25;
            else
                Cpercent=0;

    printf("La capacidad estimada de la batería en
tiempo real es: %d %\n", Cpercent);
    sprintf(send_data, "%d", Cpercent);
    send(connection, send_data, strlen(send_data), 0);

    fflush(stdout);
    break;

case 6 :

printf("Seleccionado codigo 6\n");
print_adc_channel(
calibration[NUMBER_CHANNELS][2]);

if(voltADC0>=2.128)
    Vtc=(-0.004+1.47*voltADC0);
else
    if(voltADC0>=1.704)
        Vtc=(-0.011+1.47*voltADC0);

```

```

        else
            if(voltADC0>=1.279)
                Vtc=(-0.004+1.47*voltADC0);
            else
                Vtc=(-0.005+1.47*voltADC0);
        printf("La tensión de entrada en la tarjeta TS-
7200 es: %3.2fV\n", Vtc);

        if(Vtc>4.5)
            Ip=45;
        else
            if(Vtc>3.125)
                Ip=(Vtc-2.953125)/0.034375;
            else
                if(Vtc>1.875)
                    Ip=(Vtc-2.5)/0.125;
                else
                    if(Vtc>0.5)
                        Ip=(Vtc-
2.046875)/0.034375;
                    else
                        Ip=-45;

        printf("La corriente en el primario es:
%3.2fA\n", Ip);

        sprintf(send_data, "%3.2f", Ip);
        send(connection, send_data, strlen(send_data), 0);

        fflush(stdout);
        break;

        default:    printf("El código introducido no es correcto
(Teclee q or Q para salir)\n");
                    sprintf(send_data, "El código introducido
no es correcto (Teclee q or Q para salir)");
                    send(connection,
send_data, strlen(send_data), 0);
                    fflush(stdout);
                    break;
    }

    close(sock);

}

close(devmem);

return ret_val;

```

}

### 20.3.3 Librería EP93XX\_adc.h

```
#define ADC_PAGE            0x80900000
#define ADCRESULT_OFFSET   0x0008
#define SDR_MASK            0x80000000
#define DATA_OFFSET        0x0008
#define DATA_MASK          0xFFFF //Máscara para borrar el valor anterior, de
la variable
#define ADCSWITCH_OFFSET    0x0018
#define ADC_CH0              0x0608 //Valor que puede tener el registro
ADCSwitch en función del canal
#define ADC_CH1              0x0680 //Valor que puede tener el registro
ADCSwitch en función del canal
#define ADC_CH2              0x0640 //Valor que puede tener el registro
ADCSwitch en función del canal
#define ADC_CH3              0x0620 //Valor que puede tener el registro
ADCSwitch en función del canal
#define ADC_CH4              0x0610 //Valor que puede tener el registro
ADCSwitch en función del canal
#define ADCSWLOCK_OFFSET    0x0020 //Valor que puede tener el registro
ADCSwitch en función del canal
#define UNLOCK_VAL          0xAA    //El software queda bloqueado al escribir
"AA" en el registro ADCSWLock

#define SYSCON_PAGE         0x80930000
#define ADCCLKDIV_OFFSET    0x0090
#define SYSCON_UNLOCK       0x00C0
#define TSEN_MASK           0x80000000
#define DEVICECFG_OFFSET    0x0080
#define ADCPD_MASK          0x04
#define ADCEN_MASK          0x20000

/* Declaración de funciones */
void init_ADC(unsigned long adc_page, unsigned long syscon_page);
int read_channel(unsigned long adc_page, unsigned short channel);
static char is_ADC_busy(unsigned long adc_page);

void init_ADC(unsigned long adc_page, unsigned long syscon_page)
{
    unsigned long val;

    /* Bit TSEN del registro ADClockDiv */
    //Lectura del valor del registro ADClockDiv mediante la variable val
    val = PEEK32(syscon_page + ADCCLKDIV_OFFSET);

    //Se deshabilita el software bloqueado
    POKE32(syscon_page + SYSCON_UNLOCK, UNLOCK_VAL);

    //Se pone a "1" el bit TSEN mediante la máscara correspondiente
    POKE32(syscon_page + ADCCLKDIV_OFFSET, TSEN_MASK | val);

    /* Bit DCEN del registro DeviceCfg */
    val = PEEK32(syscon_page + DEVICECFG_OFFSET);

    //Se deshabilita el software bloqueado
    POKE32(syscon_page + SYSCON_UNLOCK, UNLOCK_VAL);
}
```

```

    //Se pone a "1" el bit DCEN mediante la máscara correspondiente
    POKE32(syscon_page + DEVICECFG_OFFSET, val | ADCEN_MASK);

    /* ADCPD bit del registro DeviceCfg */
    val = PEEK32(syscon_page + DEVICECFG_OFFSET);

    //Se deshabilita el software bloqueado
    POKE32(adc_page + SYSCON_UNLOCK, UNLOCK_VAL);

    //Se pone a "0" el bit ADCPD mediante la máscara correspondiente
    POKE32(syscon_page + DEVICECFG_OFFSET, val & ~ADCPD_MASK);
}

int read_channel(unsigned long adc_page, unsigned short channel)
{
    unsigned long val;
    //Se deshabilita el software bloqueado
    POKE32(adc_page + ADCSWLOCK_OFFSET, UNLOCK_VAL);

    //Escribir en el registro ADCSwitch para seleccionar el canal
    POKE32(adc_page + ADCSWITCH_OFFSET, channel);

    /* Sondeo del registro ADCResult */

    while(is_ADC_busy(adc_page));

    // Leer el valor del registro TSXYResult register introduciendolo en la
variable val
    val = PEEK32(adc_page + DATA_OFFSET);

    // Borrar la variable local, mediante la máscara "FFFF", para "limpiar"
el registro
    val = val & DATA_MASK;

    return val;
}

static char is_ADC_busy(unsigned long adc_page)
{
    unsigned long val;

    //Lectura del registro ADCResult introduciendolo en la variable val
    val = PEEK32(adc_page + ADCRESULT_OFFSET);

    // Continuar leyendo el registro mientras el bit SDR esté a "1"
    // El bit SDR está a "1" cuando un nuevo valor aparece en el registro
    // y está a "0" cuando el valor es leído
    if((val & SDR_MASK) == SDR_MASK)
        return TRUE;

    return FALSE;
}

```

## 20.3.4 Librería peekpoke.h

```

//PEEK-Se utiliza para leer el contenido de una celda de memoria
//en una determinada dirección. Se trata de una asignación de
//registros en la memoria hardware.

```



```
//POKE-Establece el contenido de una celda de memoria

//La instrucción asm evita tener que conocer las posiciones de memoria
//que contienen los datos a los que se desea acceder.
//En el modificador "r" la variable reside en un registro
//En el modificador "memory" la variable es una posición de memoria

//La libreria contiene los PEEK-POKE de 8,16 y 32 bits
#define TRUE 0x01
#define FALSE 0x00

static inline unsigned short PEEK16(unsigned long addr) {
    unsigned short ret;

    asm volatile (
        "ldrh %0, [ %1 ]\n"
        : "=r" (ret)
        : "r" (addr)
        : "memory"
    );
    return ret;
}

static inline void POKE16(unsigned long addr, unsigned short dat) {
    asm volatile (
        "strh %1, [ %0 ]\n"
        :
        : "r" (addr), "r" (dat)
        : "memory"
    );
}

static inline unsigned long PEEK32(unsigned long addr) {
    unsigned long ret;

    asm volatile (
        "ldr %0, [ %1 ]\n"
        : "=r" (ret)
        : "r" (addr)
        : "memory"
    );
    return ret;
}

static inline void POKE32(unsigned long addr, unsigned long dat) {
    asm volatile (
        "str %1, [ %0 ]\n"
        :
        : "r" (addr), "r" (dat)
        : "memory"
    );
}

static inline unsigned char PEEK8(unsigned long addr) {
    unsigned char ret;

    asm volatile (
        "ldrb %0, [ %1 ]\n"
        : "=r" (ret)
        : "r" (addr)
        : "memory"
    );
}
```

```
        );  
        return ret;  
    }  
  
    static inline void POKE8(unsigned long addr, unsigned char dat) {  
        asm volatile (  
            "strb %1, [ %0 ]\n"  
            :  
            : "r" (addr), "r" (dat)  
            : "memory"  
        );  
    }
```



**21 INDICES**



## 21.1 ÍNDICE DE ILUSTRACIONES

Ilustración 1: Número de muertos en accidente de tráfico en España entre 1992 y el 2008 -----	11
Ilustración 2: Muertos por millón de población en el 2003-----	12
Ilustración 3: Muertos por millón de población en el 2008-----	12
Ilustración 4: Ejemplo ilustrativo del asistente de cambio de carril -----	14
Ilustración 5: Funcionamiento del ESC de Bosch (ESP)-----	16
Ilustración 6: Pantalla de BMV Night Vision -----	16
Ilustración 7: Logotipo -----	19
Ilustración 8: IVVI 2.0 -----	19
Ilustración 9: Retrovisor -----	19
Ilustración 10: Salpicadero -----	20
Ilustración 11: Maletero IVVI 2.0-----	20
Ilustración 12: Proceso para el diseño eficaz-----	21
Ilustración 13: Arquitectura del diseño -----	22
Ilustración 14: Componentes de una grupo -----	23
Ilustración 15: Detalle de funcionamiento de una batería-----	23
Ilustración 16: Detalle de los componentes de una batería-----	24
Ilustración 17: Estados de una batería -----	24
Ilustración 18: Gráfica de rendimiento de una batería de plomo ácido con respecto a la temperatura en °C -----	26
Ilustración 19: Variación del voltaje a 25 °C-----	28
Ilustración 20: Variación del voltaje a 1 °C -----	28
Ilustración 21: Diagrama del sensor de red -----	30
Ilustración 22: Diagrama de bloques del acondicionamiento de la señal de entrada-----	31
Ilustración 23: Circuito de simulación del transformador -----	33
Ilustración 24: Relación de transformación del transformador aislado -----	34
Ilustración 25: Esquema eléctrico del sensor detector de red-----	36
Ilustración 26: Tiempo de respuesta en función del valor de $R_c$ -----	37
Ilustración 27: Tiempo de respuesta para $R_c=47K\Omega$ -----	37
Ilustración 28: Tensión de rizado a la salida del condensador $R_c=47K\Omega$ -----	38
Ilustración 29: Picos de corriente en la desconexión -----	38
Ilustración 30: Comparador no inversor -----	39
Ilustración 31: Esquema eléctrico para la polarización de un zener -----	40
Ilustración 32: Circuito eléctrico completo-----	42
Ilustración 33: Acondicionamiento de la entrada de la tarjeta TS-7200-----	42
Ilustración 34: Efecto Hall-----	43
Ilustración 35: Transductor -----	44
Ilustración 36: Transductor -----	44
Ilustración 37: Esquema básico de un transductor LTS-----	45
Ilustración 38 Esquema del transductor de corriente -----	46
Ilustración 39: Relación entre la entrada y la salida del transductor de corriente -----	46
Ilustración 40: Esquema eléctrico sensor de corriente -----	49
Ilustración 41: Relación entre la tensión de salida del transductor y la tensión de salida del sensor -----	50
Ilustración 42: Curvas de funcionamiento del transductor y del sensor de corriente-----	51
Ilustración 43: Splines de modelado -----	51
Ilustración 44: Extracto de la programación en C de la curva del transductor-----	52
Ilustración 45 -----	53
Ilustración 46: Extracto de la programación de la relación entre-----	54
Ilustración 47: Resistencia adicional-----	54
Ilustración 48: Circuito medidor de la tensión en bornes de la batería adicional -----	55
Ilustración 49: Relación entre la entrada y la salida del circuito medidor de la tensión en bornes de la batería adicional-----	56
Ilustración 50: Resistencia acondicionadora del ADC4-----	57
Ilustración 51: SAI -----	59
Ilustración 52: Integración del Inversor en la arquitectura-----	63
Ilustración 53: placa TS-7200 -----	66
Ilustración 54: Esquema TS-7200 -----	66
Ilustración 55: EP9302 Cirrus Diagrama de bloques -----	68
Ilustración 56: conector tarjeta SCF -----	72

Ilustración 57: Configuración de pines del cabezal del puerto serie COM2	73
Ilustración 58: Configuración de pines del cabezal DIO1	81
Ilustración 59: Registro DDR	81
Ilustración 60: Relación entre los puertos y los GPIO	83
Ilustración 61: Señales de conexión de la lógica de control del puerto B	85
Ilustración 62: Registro PBDR dirección 0x8084_0014	85
Ilustración 63: Registro PBDR dirección 0x8084_0004 con posibilidad de lectura/escritura	86
Ilustración 64: Función de	88
Ilustración 65: Función de transferencia exacta del	88
Ilustración 66: Registro ADCSWLock en la dirección 0x8990_0020	89
Ilustración 67: Registro ADCClkDiv mapa de memoria 0x8093_0090	90
Ilustración 68: Registro DeviceCfg en la dirección de memoria 0x8093_0080	90
Ilustración 69: Registro ADCSwitch en la dirección 0x8090_0018	91
Ilustración 70: Registro ADCResult 0x8090_0008	91
Ilustración 71: Registro PEDR 0x8084_0020	92
Ilustración 72: Botón de reset en la placa TS-7200	93
Ilustración 73: Esquemático de conexiones	96
Ilustración 74: Logotipo Eclipse	97
Ilustración 75: Ventana de trabajo Eclipse	98
Ilustración 76: Configurar pantalla	98
Ilustración 77: Configuración de la ventana de trabajo	99
Ilustración 78: Ventanas más importantes	99
Ilustración 79: Creación nuevo proyecto	100
Ilustración 80: Ventana nuevo proyecto	101
Ilustración 81: Ejemplo creación nuevo proyecto	101
Ilustración 82: Completar con el nombre del autor es opcional	101
Ilustración 83: Selección del botón Advanced settings	102
Ilustración 84: Selección Internal builder/Debug	102
Ilustración 85: Selección Internal builder/Release	103
Ilustración 86: Pasos para obtener un ejecutable desde el código fuente .C	103
Ilustración 87: Compiler invocation command Debug	104
Ilustración 88: Compiler invocation command Release	104
Ilustración 89: GCC C Compiler Debug	105
Ilustración 90: GCC C Linker Debug	105
Ilustración 91: GCC Assembler Debug	106
Ilustración 92: GCC C Compiler Release	106
Ilustración 93: GCC C Linker Release	107
Ilustración 94: GCC Assembler Release	107
Ilustración 95: Ventana Select Configuration	108
Ilustración 96: Nuevo proyecto creado	108
Ilustración 97: Detalle del nuevo proyecto	109
Ilustración 98: Código que aparece por defecto	109
Ilustración 99: Código Ejemplo	109
Ilustración 100: Guardar el proyecto	110
Ilustración 101: Compilar	110
Ilustración 102: Mensaje de compilación de código sin errores	110
Ilustración 103: Detalle de nuevo proyecto compilado	111
Ilustración 104: Logotipo Fillezilla	112
Ilustración 105: Entorno Fillezilla	112
Ilustración 106: Localización	113
Ilustración 107: Localización - carpeta Release	114
Ilustración 108: Gestor de sitios	115
Ilustración 109: Ventana Gestor de sitios	115
Ilustración 110: Como completar la ventana Gestor de sitios	115
Ilustración 111: Mensajes de Fillezilla	116
Ilustración 112: Transferencia de archivos	116
Ilustración 113: Módulos de Fillezilla	117
Ilustración 114: Detalle 1 transferencia de archivos	117
Ilustración 115: Detalle 2 transferencia de archivos	117
Ilustración 116: Diagrama de bloques en la comunicación TCP/IP	119
Ilustración 117: Cambio de formato en la transmisión de datos	120

Ilustración 118: Mensajes en la consola al inicio de la comunicación	122
Ilustración 119: Carpeta EasyBattery	123
Ilustración 120: Entorno de EasyBattery	124
Ilustración 121: Mensaje error de conexión al pulsar el botón "Comprobar Conexión de red"	124
Ilustración 122: Mensaje error de conexión al pulsar el botón "Comprobar intensidad demanda"	124
Ilustración 123: Mensaje de error de conexión al pulsar el botón "Cerrar Comunicación"	125
Ilustración 124: Mensaje de conexión	125
Ilustración 125: Mensaje error sin conexión	125
Ilustración 126: Mensaje aviso sin conexión	125
Ilustración 127: Mensaje Cerrar Comunicación	125
Ilustración 128: Ventana aviso del botón: "Comprobar Conexión de red" que indica que el Sistema no está conectado a la red.	126
Ilustración 129: Ventana aviso del botón: "Comprobar Conexión de red" que indica que el Sistema está conectado a la red.	126
Ilustración 130: Ventana "Comprobando intensidad demandada"	126
Ilustración 131: Ventana: Selección de la capacidad	127
Ilustración 132: Visualización de datos cuando hay conexión con la tarjeta TS-7200	127
Ilustración 133: Eliminar datos de la Pestaña Impresión de Datos	128
Ilustración 134: Ventana Guardar como	128
Ilustración 135: Ventana Vista previa	128
Ilustración 136: Ventana Configurar página	129
Ilustración 137: Ventana Imprimir	129
Ilustración 138: Acceso a Capacidad estimada	129
Ilustración 139: Porcentajes de la capacidad de la batería	130
Ilustración 140: Mensaje error de conexión al pulsar el botón "Capacidad estimada"	130
Ilustración 141: Ventana Color	130
Ilustración 142: Añadir la Fecha a la ventana principal	131
Ilustración 143: Ventana Acerca de EasyBattery	131

## 21.2 ÍNDICE DE TABLAS

Tabla 1: Características de una batería de plomo ácido. Temperatura y vida útil	25
Tabla 2: Características de una batería de plomo ácido. Temperatura de congelación	26
Tabla 3: Características de una batería de plomo ácido. Temperatura y rendimiento	26
Tabla 4: Relación entre la capacidad y tensión en bornes de una batería de vehículo	28
Tabla 5: Valores de inductancias	33
Tabla 6: Comparativa de transformadores	34
Tabla 7: Comparación entre los modelos S20WB60 y GBPC2506	35
Tabla 8: Características del LTS-15NP	47
Tabla 9: Tabla proporcionada por el fabricante del LTS 15NP	48
Tabla 10: Relación entre la tensión de salida del transductor y la tensión de salida del sensor después del acondicionamiento	50
Tabla 11: Relación de unidades entre el transductor de corriente la tensión en el sensor de corriente y la corriente demandada por la batería	51
Tabla 12: Relación entre la tensión de entrada y la de salida del circuito medidor de la tensión en bornes	56
Tabla 13: Rango de tensiones en bornes de una batería en relación a la capacidad de la batería	56
Tabla 14: Tabla del cálculo de potencias	61
Tabla 15: Tabla de funcionamiento de la SAI	62
Tabla 16: Registros de control del watchdog	69
Tabla 17: Registro Watchdog para el tiempo de espera	70
Tabla 18: Configuración de pines del cabezal LCD	70
Tabla 19: Mapa memoria LCD	71
Tabla 20: Configuración de los pines del conector del puerto serie COM1	72
Tabla 21: Configuración de los pines del conector del puerto serie COM2	73
Tabla 22: Tabla AND	76
Tabla 23: Tabla OR	77
Tabla 24: Tabla XOR	77
Tabla 25: Comandos Linux de utilidad	80



<b>Tabla 26: Direcciones de memoria de los pines DIO</b>	<b>82</b>
<b>Tabla 27: Tamaño de bits de los puertos</b>	<b>83</b>
<b>Tabla 28: Registros adicionales de los puertos A, B y F</b>	<b>84</b>
<b>Tabla 29: Pines del cabezal DIO1 correspondiente a los SPI</b>	<b>86</b>
<b>Tabla 30: Valores de selección del canal del ADC</b>	<b>91</b>
<b>Tabla 31: Mapa de memoria LEDs</b>	<b>92</b>
<b>Tabla 32: Listado de jumpers</b>	<b>92</b>
<b>Tabla 33: Estado de los jumpers</b>	<b>93</b>
<b>Tabla 34: Relación de entradas y salidas</b>	<b>95</b>
<b>Tabla 35: Tabla de solicitudes del programa cliente</b>	<b>122</b>
<b>Tabla 36: Tabla de funcionalidades del programa servidor en función de la solicitud requerida por el programa cliente</b>	<b>122</b>
<b>Tabla 37: Presupuesto</b>	<b>133</b>

## 21.3 ÍNDICE ALFABÉTICO

### A

ABS, 12, 14  
 ACC, 12, 13, 17  
*accionadores de protección estructural*, 16  
 ADC Switch, 95, 96  
 Alerta de velocidad, 12, 16  
 Alimentación de la TS-7200, 99  
 apagar la tarjeta, 122  
 ARM, 67  
 Asistente de cambio de carril, 12, 13  
 autodescarga, 27

### B

batería, 21, 23, 24, 25, 26, 27, 39, 41, 43, 45, 46, 47, 48, 62, 67  
     tiempo de operación, 27  
*bits reservados*, 85  
 bus SPI, 88

### C

*ciclo de carga y descarga*, 25  
 COM1, 69, 74  
 COM2, 69, 74, 75  
 Complementar bits, 79  
 configuración del cabezal DIO, 83  
 Control de cruce adaptativo, 12, 13  
 Control electrónico de estabilidad, 12, 14  
*corriente de carga*, 25, 27  
 corriente de descarga, 27  
     *curva de. Véase*  
 CPLD, 71, 72  
*curva de corriente de carga*, 27

### D

Detección del ángulo muerto, 12  
 DGT, 10  
 DIO, 69, 70, 73, 80, 81, 83, 84  
     cabezal, 83  
 dirección de la tarjeta, 119

### E

EasyBattery, 127  
 EBS, 12, 14  
 eCall, 12, 14, 197  
 EEPROM, 71, 92  
 EP9301, 69  
 EP9302, 67, 69  
*eSafety Forum*, 10, 11  
 ESC, 12, 14  
 estado de un bit, 78  
 Ethernet  
     conector Ethernet, 73

### F

filtro paso bajo, 39  
 FIQ, 70

Forzar bits a 1, 79

### G

getpagesize(), 81  
 Glue Logic CPLD, 71  
 GPIO, 84, 85, 86, 198  
     enhanced, 85

### I

interrupción, 67, 70  
 IRQ, 70

### J

JTAG, 71, 72  
 jumpers, 72, 96

### L

LCD, 68, 69, 72  
 LDWS, 12, 15  
 LED, 62, 73, 81, 96  
 LEDs, 95, 96  
 Luces autoadaptables, 12, 13

### M

makefile, 105, 199  
 MAP\_PRIVATE, 82  
 MAP\_SHARE, 82  
 mapa de memoria, 71, 81, 145  
 máscaras, 77  
 matriz alfanumérica, 68, 69  
 mmap(), 72  
 MMU, 68, 70, 71

### P

PC 104, 73  
 PLD, 71  
 PROT\_READ, 82  
 puerto B, 84, 86  
 Puerto Ethernet, 68, Véase Ethernet  
 puerto serie, 69, 74, 75

### R

**read-only**, 67  
 rectificación de onda completa, 35  
 registro, 71, 72, 78, 79, 80

### S

SA, 131  
 SAI, 59, 60, 61, 62  
 SAI off-line, 59  
 SB, 131  
 SC, 131  
 SDRAM, 71  
 Sistema anti bloqueo de frenos, 12

Sistema de control de la presión en los neumáticos, 12  
Sistema electrónico de asistencia a la frenada, 12, 14  
Sistemas de control y alerta antisueño, 14  
**sistemas embebidos**, 67  
SPI, 68, 88  
STI, 11

## **T**

tiempo de carga, 27  
**Tiempo de Espera**, 72, 73, 84  
TPMS, 12  
TS-7200, 67  
TSXYResult register, 93

## **U**

UPS, 59, 61, 62, 63, 198

## **V**

Visión nocturna, 15

## **W**

Watchdog Timer, 71  
WDT, 71, 72

## **X**

Xiling, 71

## 22 LISTADO DE REFERENCIAS

- [1] [http://www.dgt.es/portal/es/seguridad\\_vial/estadistica/accidentes\\_24horas/evolucion\\_n\\_victimas/](http://www.dgt.es/portal/es/seguridad_vial/estadistica/accidentes_24horas/evolucion_n_victimas/) (19/11/11)
- [2] [http://ec.europa.eu/information\\_society/activities/intelligentcar/esafety/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/esafety/index_en.htm) (19/11/11)
- [3] [http://www.esafetysupport.org/download/press\\_releases/Microsoft%20Word%20-%20Press%20release%20eSafety%20190608\\_%20final\\_2\\_.pdf](http://www.esafetysupport.org/download/press_releases/Microsoft%20Word%20-%20Press%20release%20eSafety%20190608_%20final_2_.pdf) (19/11/11)
- [4] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_07/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_07/index_en.htm) (19/11/11)
- [5] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/benefits/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/benefits/index_es.htm) (19/11/11)
- [6] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/index_en.htm) (19/11/11)
- [7] [http://es.wikipedia.org/wiki/Antilock\\_Brake\\_System](http://es.wikipedia.org/wiki/Antilock_Brake_System) (19/11/11)
- [8] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_01/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_01/index_en.htm) (19/11/11)
- [9] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_02/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_02/index_en.htm) (19/11/11)
- [10] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_03/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_03/index_es.htm) (19/11/11)
- [11] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_04/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_04/index_es.htm) (19/11/11)
- [12] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_05/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_05/index_es.htm) (19/11/11)
- [13] <http://www.circulaseguro.com/vehiculos-y-tecnologia/tendremos-el-e-call-de-serie-por-fin> (19/11/11)
- [14] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_07/index\\_es.ht](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_07/index_es.ht) (19/11/11)
- [15] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_08/index\\_en.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_08/index_en.htm) (19/11/11)
- [16] [http://es.wikipedia.org/wiki/Antilock\\_Brake\\_System](http://es.wikipedia.org/wiki/Antilock_Brake_System) (19/11/11)
- [17] [http://www.bosch-esencial.com/us/language1/from\\_abs\\_to\\_esp.html](http://www.bosch-esencial.com/us/language1/from_abs_to_esp.html) (19/11/11)
- [18] [http://es.wikipedia.org/wiki/Control\\_de\\_estabilidad](http://es.wikipedia.org/wiki/Control_de_estabilidad) (19/11/11)
- [20] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_13/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_13/index_es.htm) (19/11/11)
- [20] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_15/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_15/index_es.htm) (20/11/11)
- [21] <http://www.mundoautomotor.com.ar/web/2007/08/05/vision-nocturna-en-bmw/> (20/11/11)
- [22] [http://www.bmw.es/es/es/insights/technology/connecteddrive/2010/safety/vision\\_assistance/night\\_vision.html#more](http://www.bmw.es/es/es/insights/technology/connecteddrive/2010/safety/vision_assistance/night_vision.html#more) (20/11/11)
- [23] [http://www.volkswagenag.com/content/vwcorp/info\\_center/en/themes/2010/06/AKTIV\\_research\\_project.html](http://www.volkswagenag.com/content/vwcorp/info_center/en/themes/2010/06/AKTIV_research_project.html) (20/11/11)
- [24] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_17/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_17/index_es.htm) (20/11/11)
- [25] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/tech\\_18/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/tech_18/index_es.htm) (20/11/11)
- [26] [http://europa.eu/legislation\\_summaries/information\\_society/other\\_policies/l31103\\_es.htm](http://europa.eu/legislation_summaries/information_society/other_policies/l31103_es.htm) (20/11/11)
- [27] [http://ec.europa.eu/information\\_society/activities/intelligentcar/technologies/benefits/index\\_es.htm](http://ec.europa.eu/information_society/activities/intelligentcar/technologies/benefits/index_es.htm) (20/11/11)
- [28] <http://europa.eu/rapid/pressReleasesActionWap.do?reference=IP/06/191&format=HTML&aged=1&language=ES> (20/11/11)
- [29] <http://www.nextautomotivesafety.com/seguridad.html#> (20/11/11)
- [30] <http://www.ptgrey.com/> (21/11/11)
- [31] <http://automotive.blogs.ie.edu/2010/03/10/%C2%BFfuncionarian-en-after-market-los-nuevos-sistemas-adas/> (21/11/11)
- [32] [http://portal.uc3m.es/portal/page/portal/dpto\\_ing\\_sistemas\\_automatica/investigacion/lab\\_sist\\_inteligentes/sis](http://portal.uc3m.es/portal/page/portal/dpto_ing_sistemas_automatica/investigacion/lab_sist_inteligentes/sis)

- [int\\_transporte/vehiculos/IvvI20](#) (21/11/11)
- [33] <http://www.embeddedarm.com/products/board-detail.php?product=TS-7200> (21/11/11)
- [34] [http://www.consumer.es/web/es/motor/mantenimiento\\_automovil/2006/05/10/151851.php](http://www.consumer.es/web/es/motor/mantenimiento_automovil/2006/05/10/151851.php) (21/11/11)
- [35] <http://yachtmen.blogspot.com/2007/05/bateria-de-plomo-acido.html> (21/11/11)
- [36] <http://www.apple.com/es/batteries/ipad.html> (22/11/11)
- [37] [http://www.engineeringtoolbox.com/lead-acid-battery-d\\_1544.html](http://www.engineeringtoolbox.com/lead-acid-battery-d_1544.html) (22/11/11)
- [38] <http://www.enalmex.com/docpdf/libro/ch05.pdf.pdf> (22/11/11)
- [39] <http://www.embeddedarm.com/documentation/ts-7200-manual.pdf> (22/11/11)
- [40] <http://www.cadence.com/products/orcad/pages/default.aspx> (22/11/11)
- [41] <http://www.seas.upenn.edu/~jan/spice/PSpicePrimer.pdf> (22/11/11)
- [42] Dato experimental facilitado telefónicamente por Saber S.L. Datos de contacto: Tel.: (93) 309 56 53 <http://www.sabersl.com/> (23/11/11)
- [43] Transformador Myrra: <http://es.farnell.com/myrra/44235/transformador-5va-230v-2-x-6v/dp/121459> (23/11/11)
- [44] Transformador Saber: <http://www.ariston.es/esp/detalle.aspx?id=1990> (23/11/11)
- [45] <http://www.electronics-lab.com/downloads/schematic/013/tutorial/PSPICREF.pdf> (23/11/11)
- [46] [http://www.lyr-ing.com/DocumentosLyR/HallEffSensors/Applications\\_of\\_LEMTransducers.pdf](http://www.lyr-ing.com/DocumentosLyR/HallEffSensors/Applications_of_LEMTransducers.pdf) (23/11/11)
- [47] [http://www.lem.com/docs/products/lts\\_15-np.pdf](http://www.lem.com/docs/products/lts_15-np.pdf) (24/11/11)
- [48] <http://www.hard-h2o.com/vertema/72408/como-seleccionar-un-sai-para-pc.html> (26/11/11)
- [49] <http://es.wikipedia.org/wiki/AVR> (26/11/11)
- [50] <http://es.wikipedia.org/wiki/Triac> (26/11/11)
- [51] [http://www.soyntec.com/es-es/item/sekury\\_a600](http://www.soyntec.com/es-es/item/sekury_a600) (26/11/11)
- [52] Información obtenida de la guía de usuario que incluyen el APC Back-UPS ES 700VA 230V
- [53] <http://www.cirrus.com/en/products/ep9302.html> (26/11/11)
- [54] <http://www.arm.com/products/processors/classic/arm9/> (26/11/11)
- [55] [http://en.wikipedia.org/wiki/ARM\\_architecture](http://en.wikipedia.org/wiki/ARM_architecture) (26/11/11)
- [56] <http://www.embeddedarm.com/products/board-detail.php?product=KIT-ARM> (26/11/11)
- [57] [http://www.embeddedarm.com/documentation/third-party/ts-7000\\_ep9302-ds.pdf](http://www.embeddedarm.com/documentation/third-party/ts-7000_ep9302-ds.pdf) (26/11/11)
- [58] [http://www.embeddedarm.com/documentation/third-party/ts-7000\\_ep9301-ug.pdf](http://www.embeddedarm.com/documentation/third-party/ts-7000_ep9301-ug.pdf) (26/11/11)
- [59] <http://tech.groups.yahoo.com/group/ts-7000/message/214> (26/11/11)
- [60] TS-ARM User's Guide, es una guía editada por Technologyc Systems que puede consultarse en: <http://www.embeddedarm.com/documentation/software/arm-linux-ts72xx.pdf> (26/11/11)
- [61] TS-7200 User's Manual, es una guía editada por Technologyc Systems que puede consultarse en: <http://arm.mydatex.cz/TS-7200.pdf> (26/11/11)
- [62] Para más información, página 91 del **EP9301 User's Manual** [58]
- [63] Más información sobre máscaras en :  
*Informática Industrial: programación en C*. Departamento de Sistemas Informáticos- Escuela de Ingenieros Industriales de Albacete. Universidad de Castilla la Mancha  
y en: [http://www.zator.com/Cpp/E4\\_9\\_3.htm](http://www.zator.com/Cpp/E4_9_3.htm) (28/11/11)
- [64] Página 523 del User's Guide EP9301 consultar referencia [61]
- [65] Para más información consultar, a cerca de estos y otros registros especiales GPIO consultar página 525 del *User's Guide EP9301* consultar referencia [61]
- [66] *Principios y aplicaciones digitales*, Albert Paul Malvino, Leach, Donald P. Leach. Ed. Marcombo. Consultar página 314.
- [67] Para información sobre este u otros puertos consultar el *User's Guide EP9301* entre las páginas 526-529, consultar referencia [61]
- [68] <http://www.wikiteka.com/trabajos/comunicacion-entre-microcontroladores-bus-spi/> (28/11/11)
- [69] <http://www.national.com/an/AN/AN-452.pdf> (28/11/11)

- [70] Para más información consultar página 20 del **TS-7200 Hardware manual**, [39] y el capítulo 19 del **EP9301 User's Guide EP9301**, que trata acerca del puerto serie síncrono del EP9302 (página 497 de la referencia [58])
- [71] Según REAL DECRETO 842/ 2002, de 2 de agosto ( B.O.E 18.09.2002) *Reglamento electrotécnico para baja tensión*, artículo 4 sección 2.b). Se puede consultar en:  
<http://www.sttmadrid.es/files-download/Reglamento%20electrot%C3%A9cnico%20baja%20tensi%C3%B3n%20%28REBT%29.pdf>  
(28/11/11)
- [72] [http://www.datasheetcatalog.net/es/datasheets\\_pdf/B/Z/X/8/BZX85C4V3.shtml](http://www.datasheetcatalog.net/es/datasheets_pdf/B/Z/X/8/BZX85C4V3.shtml) (30/11/11)
- [73] <http://www.national.com/ds/LM/LM124.pdf> (30/11/11)
- [78] La *suavización por esplines* consiste en la unión de varios segmentos hasta conformar una curva. Consultar: *Modelamiento estadístico* Jorge Ortiz Alvaro Montenegro, página 116. Universidad Nacional de Colombia.
- [79] <http://www.datasheetcatalog.org/datasheet/Shindengen/mXqwqzy.pdf> (04/12/11)
- [80] [http://www.datasheetcatalog.com/datasheets\\_pdf/G/B/P/C/GBPC2506.shtml](http://www.datasheetcatalog.com/datasheets_pdf/G/B/P/C/GBPC2506.shtml) (04/12/11)
- [81] *Electronica Analogica* José Espí Lopez et Editorial Prentice Hall (2006)
- [82] Simulado el circuito se ha comprobado que transcurrido 20 segundos el condensador aún no se ha descargado.
- [83] <http://pdfserv.maxim-ic.com/en/ds/MAX197.pdf> (05/12/11)
- [84] manual *Linux for ARM on TS-72XX* [86], páginas 39 y 40 .
- [85] Apartado 5.1 del *TS-7200 User's Manual*[61] denominado *Single Sample Acquisition Procedure*
- [86] *Linux for ARM on TS-72XX User's Guide*- Technologic Systems  
<http://www.embeddedarm.com/documentation/software/arm-linux-ts72xx.pdf> (05/12/11)
- [87] Para ver los valores del resto de canales o más información de este registro ver página 520 del *EP9301 User's Manual* [61]
- [88] En el *TS-7200 Hardware manual* [39] confunde el registro.
- [89] En la página 28 del *TS-7200 Hardware manual* [39].
- [90] Existen ejemplos de disponibles sobre el funcionamiento del ADC en el servidor ftp:  
<ftp://ftp.embeddedarm.com/ts-arm-sbc/ts-7200-linux/samples/> (05/12/11)
- [91] <http://www.eclipse.org/> (07/12/11)
- [92] <http://es.wikipedia.org/wiki/PuTTY> (07/12/11)
- [93] <http://busybox.net/about.html> (07/12/11)
- [94] <http://www.embeddedarm.com/software/solutions-arm.php> (07/12/11)
- [95] <http://www.debian.org/index.es.html> (07/12/11)
- [96] <http://filezilla-project.org/> (07/12/11)
- [97] [http://alejandria.nidaval.com/scripts/Editorial.dll?SE=2\\_1\\_0\\_T4\\_A168\\_175](http://alejandria.nidaval.com/scripts/Editorial.dll?SE=2_1_0_T4_A168_175) (08/12/11)
- [98] <http://profesores.elo.utfsm.cl/~agv/elo320/makefile/makefile.html> (08/12/11)
- [99] <http://www.chuidiang.com/clinix/herramientas/makefile.php> (08/12/11)
- [100] <http://iie.fing.edu.uy/~vagonbar/gcc-make/gcc.htm> (08/12/11)
- [101] [http://coe.uncc.edu/~jmconrad/GradStudents/Thesis\\_Thakur.pdf](http://coe.uncc.edu/~jmconrad/GradStudents/Thesis_Thakur.pdf) (08/12/11)
- [102] <http://www.tenouk.com/ModuleW.html> (08/12/11)
- [103] <http://www.filezilla.com.ar/> (08/12/11)
- [104] <http://nosetup.webcindario.com/filezilla/> (08/12/11)
- [105] ver página 13 del *TS-7200 Hardware Manual* [39]
- [106] <http://ideaslibres.wordpress.com/2008/08/18/instalar-y-configurar-eclipse-para-programar-en-python/> (09/12/11)
- [107] <http://www.eclipse.org/downloads/packages/release/europa/winter> (09/12/11)
- [108] Para más información sobre su adaptación a Windows consulte:  
<http://mundogeek.net/archivos/2007/01/17/desarrollo-con-cc-en-eclipse/> (09/12/11)
- [109] <http://www.it.uniovi.es/docencia/GestionGijon/redes/Redes-Practica1-4.pdf> (15/12/11)

- [110] <http://www.fic.udc.es/files/asignaturas/56XR/files/ManualSocketsC-2009.pdf> (16/12/11)
- [111] <http://es.wikipedia.org/wiki/UPnP> (18/12/11)
- [112] <http://www.enalmex.com/docpdf/libro/ch05.pdf.pdf> (02/01/11)
- [113] <http://www.electronics-lab.com/downloads/schematic/013/tutorial/PSPCREf.pdf>  
(05/01/11)
- [114] [http://www.electroindustrial.com/inversores\\_12v\\_a\\_220v/whs\\_600\\_12.htm](http://www.electroindustrial.com/inversores_12v_a_220v/whs_600_12.htm) (09/01/12)